

Computer Treatment of Large Air Pollution Models

Environmental Science and Technology Library

VOLUME 2

The titles published in this series are listed at the end of this volume.

Computer Treatment of Large Air Pollution Models

by

Zahari Zlatev

*National Environmental Research Institute,
Roskilde, Denmark*



SPRINGER-SCIENCE+BUSINESS MEDIA, B.V.

Library of Congress Cataloging-in-Publication Data

Zlatev, Zahari,

Computer treatment of large air pollution models / by Zahari Zlatev.

p. cm. -- (Environmental science and technology ; v. 2)

Includes bibliographical references (p.) and indexes.

ISBN 978-94-010-4137-9 ISBN 978-94-011-0311-4 (eBook)

DOI 10.1007/978-94-011-0311-4

1. Air--Pollution--Computer simulation. I. Title. II. Series:
Environmental science and technology (Dordrecht, Netherlands) ; v.
2.

TD883.1.Z58 1995

628.5'3'0113--dc20

94-45303

ISBN 978-94-010-4137-9

Printed on acid-free paper

All Rights Reserved

© 1995 Springer Science+Business Media Dordrecht

Originally published by Kluwer Academic Publishers in 1995

Softcover reprint of the hardcover 1st edition 1995

No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the copyright owner.

To my wife

Ida

and my daughters

Anna Irina and Tanja

TABLE OF CONTENTS

Preface xi

Acknowledgement xv

1. The Air Pollution Problem 1

1.1. Long-range transport of air pollution 2

1.2. Emission sources 3

1.3. Transport of air pollutants 6

1.4. Diffusion phenomena 7

1.5. Chemical reactions during the transport 8

1.6. Deposition of air pollutants 10

1.7. Effects on plants, animals and humans 11

1.8. Need for reduction of the air pollution 12

1.9. Reduction of air pollution versus its economical cost 12

1.10. Critical levels, control strategies and long-range transport models 14

1.11. Concluding remarks and references concerning the air pollution
problem 19

**2. Mathematical Models for Studying the Long-Range Transport of
Air Pollutants 21**

2.1. Long-range transport and reduction of air pollution 22

2.2. Use of models versus use of measurements 26

2.3. Modelling the transport in the atmosphere 28

2.4. Modelling the diffusion phenomena 35

2.5. Modelling the deposition processes 39

2.6. Choice of a chemical scheme 41

2.7. Mathematical description of the chemical reactions 49

2.8. Introduction of the emissions in the models 52

2.9. General mathematical description of an air pollution model 53

2.10. Lagrangian models versus Eulerian models 54

2.11. Concluding remarks and references concerning large models for
studying long-range transport of air pollutants 66

3. Numerical Treatment of Large Air Pollution Models 69

3.1. Splitting of the large model to sub-models 70

3.2. Treatment of the horizontal advection 73

3.3. Algorithms for the horizontal diffusion terms 86

3.4. The chemical terms and the problem of stiffness 88

3.5. Treatment of the deposition terms 100

3.6. Initial value conditions 101

3.7. Boundary value conditions 101

3.8. Treatment of the vertical advection and diffusion 104

- 3.9. Use of local refinement in the discretization process 105
 - 3.10. Concluding remarks and references concerning the numerical treatment of large air pollution models 106
- 4. **Testing the Reliability of the Numerical Algorithms** 111
 - 4.1. Testing the advection algorithms 112
 - 4.2. Testing the numerical algorithms used in the combination of the advection and chemistry sub-models 118
 - 4.3. Other tests of numerical algorithms 129
 - 4.4. Concluding remarks and references concerning the tests of the reliability of the numerical algorithms 139
- 5. **Need for Efficient Algorithms** 141
 - 5.1. Size of the discretized models 142
 - 5.2. Efficient implementation of the ODE's algorithms 143
 - 5.3. Need for an automatic adjustment of the stepsize 154
 - 5.4. Variable stepsize variable formula methods 156
 - 5.5. Zero-stability properties of the VSVFM's 161
 - 5.6. Control of the accuracy of the results 166
 - 5.7. Designing algorithms with desired properties 168
 - 5.8. Concluding remarks and references concerning the implementation of variable stepsize variable formula methods 172
- 6. **Computations on High-Speed Computers** 175
 - 6.1. Classification of high-speed computers 176
 - 6.2. Vector processors 178
 - 6.3. Parallel computers with shared memory 182
 - 6.4. Parallel computers with distributed memory 183
 - 6.5. High-speed computers with more complicated architecture 185
 - 6.6. High-speed computers and large air pollution problems 188
 - 6.7. Concluding remarks and references concerning the efficiency of the numerical algorithms and the use of high-speed computers 189
- 7. **Running Air Pollution Models on Vector Machines** 193
 - 7.1. Separating the most time-consuming operations 193
 - 7.2. Using vectorized basic linear algebra operations 197
 - 7.3. Vectorizing matrix-vector multipliers for banded matrices 198
 - 7.4. Vectorizing the solution of systems with banded matrices 211
 - 7.5. Vectorizing fast Fourier transforms 220
 - 7.6. Concluding remarks and references concerning efficient runs of large air pollution models on vector machines 222

8.	Running Models on Parallel Machines with Shared Memory	225
8.1.	Identifying operations that can be run concurrently	226
8.2.	Calling subroutines in concurrent runs	227
8.3.	Performing both vectorization and parallelization	238
8.4.	Numerical experiments on two ALLIANT computers	244
8.5.	Concluding remarks and references concerning efficient runs on parallel machines with shared memory	248
9.	Running Models on Massively Parallel Computers	251
9.1.	Basic information on the Connection Machine	252
9.2.	Mapping large arrays into the memory of a massively parallel computer	254
9.3.	Need to reorganize some computations on a massively parallel computer	255
9.4.	Need to restructure the modules used in the models	256
9.5.	Some results obtained on the Connection Machine (CM-200)	257
9.6.	Concluding remarks and references concerning efficient runs of large air pollution models on massively parallel machines	263
10.	Numerical Experiments with the Danish Eulerian Model	265
10.1.	The position of the Danish Eulerian Model among the other LRTAP models	266
10.2.	Calculations of the levels of the concentrations and depositions of air pollutants in Europe	269
10.3.	Comparisons with measurements	272
10.4.	Studying the influence of sources in one region on the air pollution in the surrounding regions	284
10.5.	Running scenarios in which the emissions are varied	288
10.6.	Studying episodes with high ozone concentrations	298
10.7.	Studying the concentrations and depositions due to Danish sources	299
10.8.	Concluding remarks and references concerning simulations with large air pollution models	301
	References	307
	Author Index	349
	Subject Index	355

P R E F A C E

"Models are often the only way of interpreting measurements to investigate long-range transport, and this is the reason for the emphasis on them in many research programs".

B. E. A. Fisher: *"A review of the processes and models of long-range transport of air pollutants"*, Atmospheric Environment, **17**(1983), p. 1865.

Mathematical models are (potentially, at least) powerful means in the efforts to study transboundary transport of air pollutants, source-receptor relationships and efficient ways of reducing the air pollution to acceptable levels. A mathematical model is a complicated matter, the development of which is based on the use of

- (i) various mechanisms describing mathematically the physical and chemical properties of the studied phenomena,
- (ii) different mathematical tools (first and foremost, partial differential equations),
- (iii) various numerical methods,
- (iv) computers (especially, high-speed computers),
- (v) statistical approaches,
- (vi) fast and efficient visualization and animation techniques,
- (vii) fast methods for manipulation with huge sets of data (input data, intermediate data and output data).

The mathematical description of the physical and chemical properties of the air pollutants is the most important task. However, there are a lot of difficult and unresolved problems connected with the physical and chemical background of the air pollution phenomena that are of interest. Therefore many different possible descriptions of the physical and chemical properties are to be tried in the model and the results obtained are to be carefully compared and studied. It is important to emphasize here that correct conclusions could be drawn only if all problems related to the remaining issues (ii)-(vii) are successfully solved. This means that: the numerical methods should be sufficiently accurate, the computers should be sufficiently fast, the statistical approaches should be reliable, efficient visualization and animation programs should be available and it should be possible to perform quick manipulations with large data sets. It is absolutely necessary to satisfy these conditions. If these conditions are satisfied, then one will be able to perform long series of simulations with the model and to derive reliable conclusions about the way in which physics and the chemistry should be implemented.

Some important problems concerning the implementation of numerical methods in a large air pollution model and the use of high-speed computers to treat such a model are the main topic of this book. It should be emphasized here that these two factors (fast and sufficiently accurate numerical methods and efficient codes for different high-speed computers) are often

underestimated. In many cases fast but not sufficiently accurate numerical methods are selected. Simplified (and oversimplified) physical mechanisms are incorporated in the models in order to make it possible to treat them on, say, personal computers. In many important situations the use of these two approaches is not justified. The numerical methods must be sufficiently accurate in order to avoid the influence of numerical errors on the results when different physical mechanisms are tested. The use of old-fashioned sequential and scalar computers (instead of modern vector and parallel computers) may cause difficulties in the selection of the right physical mechanisms (some promising physical mechanisms have to be rejected and not included in the set of the tested mechanisms only because they are complicated and it will be impossible to run the resulting model on such slow computers).

The long-range transport of air pollutants and the major physical processes involved in it are briefly described in Chapter 1. Some problems connected with the reduction of the air pollution to acceptable levels is also discussed in Chapter 1.

The mathematical description of the major physical processes, which take place during the transport of air pollutants, is studied in Chapter 2. A general Eulerian model is derived and then compared with models based on the Lagrangian approach.

Various numerical methods that can be used in connection with large air pollution models are discussed in Chapter 3. A comparison of different numerical methods is also given there.

It is important to be able to check the accuracy of the numerical methods. It is highly desirable to implement error estimators in all parts of a large air pollution model, but it is extremely difficult to accomplish such a task. In fact, this is a very challenging but still unresolved problem. Therefore, a carefully chosen set of test-problems has to be used. Test-problems are described in Chapter 4.

In many cases the use of advanced methods, which try to adjust automatically some parameters in an attempt to achieve high performance, is worthwhile. The use of such methods during the time-integration part of the computation is demonstrated in Chapter 5.

The discretization of any large air pollution model leads to huge systems of ordinary differential equations. It is important to apply high-speed computers in the treatment of these systems. Basic properties of several classes of modern high-speed computers are discussed in Chapter 6.

The application of various vector processors in the solution of some basic algebraic operations, which appear again and again when big application codes (not only air pollution models) are handled on computers, is discussed in Section 7. The efficiency of the devices proposed there is illustrated by many numerical examples.

Parallel machines with shared memory are often used in the attempt to achieve high speeds of computations. The application of such computers in the treatment of large air pollution problems is discussed in Section 8.

Many scientists believe that the future of scientific computing is the use of powerful massively parallel computers that could have thousands of processors. An application of such

a computer, the Connection Machine, in the treatment of some modules of a large air pollution model is described in Chapter 9. The computational speed achieved is illustrated in tables and by plots.

Some experiments with the particular large air pollution model developed at the National Environmental Research Institute (Roskilde, Denmark), the Danish Eulerian Model, are described in Chapter 10. The results are illustrated by many plots.

Air pollution modelling is an interdisciplinary science. Meteorologists, physicists, applied mathematicians, numerical analysts, computer scientists, statisticians and chemists are working in this area. The main topics discussed in the book are numerical methods and high speed scientific computations. However, an attempt is made to take into account the fact that specialists from many other areas might be interested in some parts of the book. This is why the following actions were taken during the preparation of the text. Unnecessary details are avoided. The notation is simplified as much as possible. The main ideas, on which the algorithms discussed are based, are described and references, where details can be found, are given. Some popular algorithms have been published, sometimes with different interpretations, in physical, meteorological, chemical and mathematical journals. An attempt to give all these references is made. This increased the number of references. The hope is that different scientists will be able to find a reference published in a journal where the notation and the language are familiar to them.

The different chapters of this book, and even some sections, can be studied independently. Moreover, short cross-references are in general avoided in the book. That is if a few formulae or several sentences from an earlier chapter are needed then they are repeated. Further, the tables and figures are supplied with as many explanations as possible so as to allow the reader to understand their contents without searching for the page in the text where the contents of the table or the figure under consideration are explained. An attempt has been carried out to use concepts and tools that are used by all scientists involved in this field (applied mathematicians, numerical analysts, meteorologists, physicists, chemists, statisticians and computer scientists).

The terminology used in connection with air pollution studies deserves some attention. There are many different names that are associated with the air pollution models: meteorological models, dispersion models, diffusion models, numerical models, computer models, chemical models, etc. The situation is sometimes rather confusing. Therefore it seems to be useful to try to state clearly what terminology will be used in this book. The following convention rules are used in the text:

(i) In this book, a model is either an air pollution model or a mathematical model (the two names being used as synonyms in all chapters of the book). It is also assumed that the model is described mathematically by a system of partial differential equations.

(ii) The different terms in the equations reflect the particular mechanisms that are used to describe mathematically the physical processes involved in the model. A group of such terms is considered as a sub-model, whose name is the name of the physical process described by these terms;

for example, the diffusion sub-model contains the terms by which the diffusion process is described mathematically.

(iii) The mathematical model is always treated numerically. The term "numerical model" is never used in the book.

(iv) The numerical treatment of a large mathematical model is always carried out on a computer. This means that the use the term "computer model" is redundant, and this term is never used in the book.

I hope that this book will be useful for all groups of scientists who are doing research in the field of air pollution modelling (applied mathematicians, numerical analysts, physicists, chemists, meteorologists, statisticians and computer scientists). Some chapters of the book are quite general and could also be used by specialists involved in other fields of science and engineering where large-scale computations are to be carried out (perhaps on high-speed computers). Thus, this book may be used as a reference book in many scientific and engineering fields where large mathematical models are to be treated numerically (in most of the cases on high-speed computers).

A C K N O W L E D G E M E N T S

R. Berkowicz (National Environmental Research Institute, Roskilde, Denmark), **A. T. Chronopoulos** (University of Minnesota, USA), **K. Gallivan** (University of Illinois at Urbana-Champaign, Illinois, USA), **B. Neta** (Naval Postgraduate School, Monterey, California, USA) and my daughter **Tanja Zlateva** read chapters of this book. They corrected my language and made many suggestions for improvements. I should like to thank them very much.

The work on the **Danish Eulerian Model** has been carried out in a close collaboration with Norwegian scientists. I should like to thank very much **A. Eliassen**, **E. Berge**, **D. Simpson** (all three from the Norwegian Meteorological Institute), **Ø. Hov** (University of Bergen) and **T. Iversen** (University of Oslo) for long, long common work on different problems related to the development of large air pollution models.

High performance on vector and parallel architectures has been achieved in collaboration with specialists from the Danish Computing Centre for Research and Education (UNI-C). I should like to thank **B. S. Andersen**, **P. C. Hansen**, **J. Moth**, **W. Owczarz** and **J. Wasniewski** for helping me to run some modules of the **Danish Eulerian Model** on different high-speed computers.

Many of the numerical methods in the **Danish Eulerian Model** have been implemented after long discussions with numerical analysts from the Institute of Numerical Analysis (this name has recently been changed to the Institute for Mathematical Modelling) at the Technical University of Denmark. I should like to thank very much **all of them**, especially **V. A. Barker** and **P. G. Thomsen**.

I should like to thank very much **L. P. Prahm** (now Director of the Danish Meteorological Institute), who introduced me to the field of large air pollution models. Also thanks to **L. Walsted Kristiansen** and **Chr. Lohse** for giving me excellent working conditions at the Division of Emissions and Air Pollution at the National Environmental Research Institute (Roskilde, Denmark). My Ph D students **J. Christensen** and **J. Brandt** participated actively in the development of the **Danish Eulerian Model**. I should like to thank them very much. Many thanks also to **all my colleagues** from the Division of Emissions and Air Pollution, especially to the scientists from the former Section of Meteorology.

During the work on this project I received several grants from the **Danish Natural Science Research Council** and the **Danish Technical Science Research Council**. I received also support from **NMR** (the Nordic Council of Ministers), **EMEP** (European Monitoring and Evaluation Programme at the Economic Commission for Europe at the United Nations) and **NATO** (North Atlantic Treaty Organization) I should like to thank all these institutions very much for supporting my research.

CHAPTER 1

THE AIR POLLUTION PROBLEM

"The quality of the atmosphere, on which existing terrestrial forms of life are dependent, has been recognized as an important variable of the environment only during the past few decades".

Leslie A. Chambers: *"Classification and Extent of Air Pollution Problems". In: "AIR POLLUTION" (A. C. Stern, ed.), Vol. 1, p. 5. Academic Press, New York, 1976.*

The air pollution, and especially the reduction of the air pollution to an acceptable level, is an important environmental problem. The air pollution is not limited to regions where there are big emission sources. Air pollutants can be transported, mainly by advection due to the wind (but diffusion phenomena and chemical transformations under the transport also play an important role), to areas where there are no emission sources. Thus, the atmosphere is normally polluted not only in areas with emission sources, but also in the surrounding areas. Furthermore, air pollutants are deposited on the surface not only in areas where there are big emission sources, but also in the surrounding areas.

The pollutants, both when these are dispersed in the atmosphere and when these are deposited on the surface, can be dangerous for plants, animals and humans if their concentrations are over certain levels. This imposes or, at least, may impose a requirement to reduce the air pollution in some areas to certain acceptable (or critical) levels. However, it must immediately be stressed that the reduction of the air pollution is an expensive process (or, at least, very often is an expensive process). Therefore one should be very careful. It is necessary to define exactly what kind of reduction is wanted and where it is necessary to achieve reductions. The acceptable (or critical) levels of the concentrations and the depositions are to be determined precisely and the air pollution should be reduced to these levels but no more, because the extra efforts may be very expensive and, therefore, may cause great economical difficulties (for some countries, at least).

Mathematical models of air pollution phenomena (and, first and foremost, of **long range-transport** phenomena) are indispensable tools in the efforts to solve the problems sketched above. Moreover, many problems connected with an optimal reduction of the air pollution to acceptable levels can successfully be solved only if reliable mathematical air pollution models are used.

A short description of the physical nature of the air pollution transport will be presented in the first chapter. Some related issues, and especially the efforts to reduce the air pollution to some acceptable levels, will also be discussed in some more detail in this chapter. Then, in the following chapters, some mathematical air pollution models that lead to very large computing problems will be studied.

1.1. Long-range transport of air pollution

"Transport, chemical reactions and deposition of atmospheric pollutants as well as their effects like smog formation and acidification of soils constitute a regional problem".

Adolph Ebel: *"Modelling of Pollutants Dispersion over Europe: An Introduction to EUROTRAC Subproject EUMAC".* In: **"TRANSPORT AND TRANSFORMATION OF POLLUTANTS IN THE ATMOSPHERE"** (P. Borrell, P. M. Borrell and W. Seiler, eds.), pp. 227-230. SPB Academic Publishing bv, The Hague, Netherlands, 1991.

The physical phenomenon, which is well-known under the name "long-range transport of air pollution" (the abbreviation LRTAP will often be used both in this chapter and in the following chapters), consists of three major stages:

1. *Emission.* During this first stage different pollutants are emitted in the atmosphere from different emission sources. Many of the emission sources are anthropogenic¹ (the terms human-made and man-made are also commonly used), but some of the air pollutants are also emitted from natural emission sources.

2. *Transport.* The actual transport of air pollutants takes place in the second stage. The driving factor is the wind. The transport of air pollution in the atmosphere that is caused by the wind is normally called "advection of the air pollutants".

3. *Transformations during the transport.* Three major physical processes take place during the transport of pollutants in the atmosphere:

3.1. *Diffusion.* The air pollutants are widely dispersed in the atmosphere (both in the horizontal directions and in the vertical direction).

3.2. *Deposition.* Some of the pollutants are deposited to the various surfaces of the Earth (soil, water and vegetation). Two different kinds of deposition phenomena are usually considered: dry deposition and wet deposition. The dry deposition continues throughout the long-range transport, while the wet deposition process takes place only when it rains.

3.3. *Chemical reactions.* Many different chemical reactions take place during the transport of pollutants in the atmosphere. As a result of the chemical reactions many secondary pollutants are created (the air pollutants that are emitted directly from the emission sources in the atmosphere are

¹ **Anthropogenic emissions:** The use of the word "anthropogenic" in connection with emissions is etymologically not correct; see Wellburn [474], p. 2. The word anthropogenic is derived from the word anthropogenesis (which is a composition of the Greek words anthropo and genesis), and "anthropogenesis" means the study of the origin of man. Nevertheless, the term "anthropogenic emissions" is commonly used instead of emissions from human-made or man-made sources.

often called primary pollutants). Some of the chemical reactions lead to a production of final species that are not damaging (thus, the final products of the chemical reactions are not necessarily air pollutants).

1.2. Emission sources

"For many years, emission data have been reported without specific guidelines on how such data should be estimated. The first guidelines, adopted in 1988, were rather general".

Harald Dovland: *"Emission Data: Agreed Reporting Guidelines and Situation in Practice"*. In: **"EMEP WORKSHOP ON THE CONTROL OF PHOTOCHEMICAL OXIDANTS IN EUROPE"** (P. Anttila, ed.), pp. 141-144. Finnish Meteorological Institute, Sahaajankatu 22, FIN-00810 Helsinki, 1993.

It is commonly accepted to classify the air pollutants into two large groups: primary air pollutants and secondary air pollutants (this has been mentioned at the end of the previous section). The primary air pollutants are directly emitted in the atmosphere from various emission sources. The secondary air pollutants appear as a result of chemical transformations during the transport of pollutants in the atmosphere. It must be mentioned here (see also the previous section) that some of the final products of the chemical reactions can be harmless. Of course, some of the chemical reactions can be reversible (under some conditions, at least). This means that not only are primary air pollutants emitted from different emission sources, but some of them can also be produced in the atmosphere during the transport.

There are two major groups of emission sources. The primary air pollutants can be emitted from natural emission sources. For example, some organic air pollutants (from a big group which is well-known under the abbreviation VOC²) are emitted from the forests. Some sulphur compounds are emitted from the seas.

While it is true that some air pollutants are emitted from natural sources, it is unfortunately also true that many of the emission sources (at least those in the highly industrialized parts of the world; Europe, North America and some parts of Asia) are created by human activities and, thus, emitted by anthropogenic (man-made, human-made) emission sources.

A detailed description of the emission sources for various air pollutants (the terms *"species"* and *"compounds"* also being commonly used, especially in connection with the chemical reactions) is beyond the scope of this book. However, it will be illustrative to explain, by an example, how the inventories concerning the anthropogenic (man-made, human-made) emission sources are prepared in order to be used in different investigations (for instance, as input data in the large air pollution models).

²**VOC:** Volatile Organic Compounds

Let us consider, as an illustration only, the anthropogenic emissions of nitrogen oxides. The anthropogenic nitrogen oxides are mainly produced, and emitted into atmosphere, during different combustion processes. The main types of anthropogenic emission sources for nitrogen oxides, according to Lübker [294] and Lübker and Tilly [296], are listed in Table 1.1.

For the 12 OECD³ countries (Austria, Denmark, Finland, France, Germany, Italy, the Netherlands, Norway, Portugal, Sweden, Switzerland and the United Kingdom) the average percentages of the contributions from the six major types of emission sources for nitrogen oxides for 1980 are given in Table 1.2 (more details can be found, for example, in Dovland and Saltbones [120], Lübker [294], Lübker and Tilly [296], Semb and Amble [392]).

Type	Source
1	Transportation
2	Power plants
3	Domestic combustion
4	Industrial combustion
5	Other industrial processes
6	Other sources

Table 1.1
The types of emission sources
for nitrogen oxides.

In some of the OECD countries the percentages of the emissions of nitrogen oxides from different types of emission sources can deviate quite a lot from the average values for the whole region of the OECD countries. For example, there is no contribution to the emissions of nitrogen oxides from power plants in Norway and Switzerland, where the natural water resources are fully exploited. The differences in the contributions of nitrogen oxides from the main types of emission sources in three of the OECD countries are clearly seen in Table 1.3.

The above short description of the different types of emission sources for nitrogen oxides is quite sufficient for the purposes of this book. It illustrates clearly how the emissions sources of nitrogen oxides are classified and how inventories that can be used in different environmental studies can be prepared. Similar procedures can be applied when inventories concerning other emissions are to be prepared (as, for example, inventories for emissions of sulphur, ammonia, different hydro-carbons, etc.). More details about many procedures (which

³OECD: Organization for Economic Co-operation and Development

can be used in connection with emissions and inventories for emissions) can be found in Lamb et al. [279], Lübkert and Schöpp [295], Seinfeld [391] or Wellburn [474]; see also the references given above.

Type	Source	Percent
1	Transportation	53
2	Power plants	24
3	Domestic combustion	6
4	Industrial combustion	12
5	Other industrial processes	4
6	Other sources	1

Table 1.2
The contributions, in percent, to the emissions of nitrogen oxides
from the different types of sources (mean values for 1980
and for the whole OECD region).

Type	Source	Denmark	Norway	UK ⁴	OECD
1	Transportation	33	75	37	53
2	Power plants	50	0	44	24
3	Domestic combustion	9	4	3	6
4	Industrial combustion	7	12	13	12
5	Other industrial processes	0	9	0	4
6	Other sources	1	0	3	1

Table 1.3
The contributions (in percent) to the emissions
of nitrogen oxides from different types of sources
in three OECD countries (Denmark, Norway and UK)
and in the whole OECD area (the last column).

⁴ **UK:** United Kingdom

1.3. Transport of air pollutants

"The atmosphere serves both as a reservoir in which chemical and physical reactions take place and as a pathway for material to go from one geophysical reservoir to another".

Douglas M. Whepdale and Jenie L. Moody: *"Large-scale Meteorological Regimes and Transport Processes"*. In: **"THE LONG-RANGE ATMOSPHERIC TRANSPORT OF NATURAL AND CONTAMINANT SUBSTANCES"** (A. H. Knap, ed.), pp. 3-36. Kluwer Academic Publishers, Dordrecht-Boston-London, 1990.

The long-range transport of different pollutants in the atmosphere is mainly due to the wind. This is a very important physical process. The long-range transport of air pollutants in the atmosphere may cause severe damaging problems (under certain meteorological conditions, at least) even in regions where there are no big emission sources. Therefore this process should be studied very carefully.

There are two questions of great interest (or, at least, may be of great interest in certain circumstances) that have to be answered:

- (1) How big is the contribution of the emission sources from a given region to the surrounding regions?
- (2) What is the influence of the reduction of the emission sources in a given region on the pollution in the surrounding regions?

Correct answers to these two important questions are needed in the efforts to reduce the pollution to some prescribed levels in an optimal way. The reduction of the pollution is normally an expensive process. Therefore it is important to reduce the pollution as much as needed, but no more. It is clear that if reliable information about

- (i) the contributions due to sources in a given region to the pollution in the surrounding regions

and

- (ii) the influence of a given reduction of the emissions from certain sources to the concentrations of the appropriate pollutants in the surrounding regions

is available, it will be possible, in principle at least, to design some optimal (or nearly optimal) strategies for the reduction of the air pollution to an acceptable level.

It is necessary to develop reliable mathematical models in order to answer these questions (or other similar questions). Moreover, reliable mathematical models are perhaps the only tool by which one can obtain detailed information both about the contribution due to sources in a given region to the pollution in the surrounding regions and about the influence of a prescribed reduction of some sources on the concentrations of the corresponding pollutants

in the surrounding regions. The correct treatment of the transport due to the wind (this transport is also called advection; see Section 1.1) is an important requirement for such a model. However, one should remember that, while the long-range transport of pollutants in the atmosphere which is caused by the wind is without doubt one of the most important processes, the other physical processes (the processes that take place during the transport) must also be taken into account during the preparation of a mathematical model.

1.4. Diffusion phenomena

"The differential equation which has been the starting point of most mathematical treatments of diffusion from sources is a generalization of the classical equation of conduction of heat in a solid and is essentially a statement of conservation of the suspended material".

F. Pasquill and F. B. Smith: *"Atmospheric Diffusion"*. Wiley and Sons, New York, 1983, pp. 94-95.

When air pollutants are released from an emission source into the atmosphere, they are transported by the wind (see the previous section). Several physical processes take place during the transport. The diffusion phenomenon is one of these processes. The air pollutants can be diffused (dispersed) both in the two horizontal directions and in the vertical direction. There are no clear limit barriers for the diffusion in the horizontal directions. However, there are two important barriers for the diffusion in the vertical direction that must be taken into account and correctly handled in the mathematical models.

The first of the two barriers is the top of the boundary mixing layer, which is determined by a special quantity, namely the mixing height. During the night the mixing height is normally rather small (the top of the boundary layer is close to the surface). During the day the mixing height can be pretty high; up to several kilometres. The air pollutants are mixed relatively fast within the boundary mixing layer.

The second of these barriers is the tropopause (its height is about 10 km). While the air pollutants can be diffused over the top of the mixing boundary layer (although this process is normally very slow), there is practically no mixing over the tropopause (even if such mixing takes place, then it is so small and/or it happens so rarely that it can be ignored, and is often ignored, in the air pollution models).

The diffusion phenomena depend on many factors (as, for example, on the meteorological conditions and on the terrain). The correct treatment of these factors in an air pollution model becomes more and more important when the space domain of the model becomes smaller. Thus, one should be more careful in the description of the diffusion process in short-scale models.

A mathematical description of the diffusion process for the case where long-range transport of air pollutants is studied (which is based on partial differential equations) will be presented in the next chapter. The physical background of the diffusion phenomena is beyond the scope of this book (a good study of the physical background of the diffusion process can

be found in, for example, Pasquill and Smith [347]). It should be stressed here, however, that the mathematical description of the diffusion process, which will be presented in the following chapter and which is used in many large air pollution models, is based on rather crude assumptions (and there is a strong need for improvements based on a more solid physical knowledge). On the other hand, it must also be emphasized that the more complicated modules for calculating the diffusion of the concentrations will as a rule require some additional input data files. Therefore, they are useful only if the available additional input data files are sufficiently accurate.

It must also be emphasized that some artificial diffusion is always presented when the air pollution transport is modelled and, moreover, when the mathematical models are treated numerically. In this situation the spatial domain under consideration is divided into small boxes by some kind of grids and it is assumed that the emissions from all sources in any of these boxes are mixed instantaneously inside the box. This is why many modellers prefer to work without any horizontal diffusion in their mathematical models; the horizontal grid-squares are often very large (the size of the horizontal grid-squares in the EMEP⁵ models is 150 km x 150 km; see Eliassen and Saltbones [154], Iversen [251], Iversen et al. [252-254] or Simpson [413,414]). The artificial diffusion will be discussed in more detail in the following chapters where large air pollution models will be studied.

1.5. Chemical reactions during the transport

"In previous model studies, it has been concluded that choice of chemical mechanism may have a decisive influence on model performance".

R. G. Derwent and T. J. Davis: *"Modelling the Impact of NO_x Control on Photochemical Ozone in Europe"*. In: **"EMEP WORKSHOP ON THE CONTROL OF PHOTOCHEMICAL OXIDANTS IN EUROPE"** (P. Anttila, ed.), p. 55. Finnish Meteorological Institute, Sahaajankatu 22 E, FIN-00810 Helsinki, Finland, 1993.

Different chemical reactions take place during the transport of pollutants in the atmosphere. The chemical reactions may lead to the creation of other air pollutants (which will be called secondary pollutants; see Section 1.2), but the chemical reactions may also lead to the conversion of some primary pollutants to harmless compounds. Therefore the chemical reactions should be studied carefully.

Let us assume that one wants to study a large group of important air pollutants (as, for example, sulphur pollutants, nitrogen pollutants, ozone and hydro-carbons) by using a mathematical model. Then it is highly desirable to determine an optimal set of chemical reactions that has at least the following two properties:

- (a) It describes adequately the reality (i.e. the actual chemical transformations that take place in the atmosphere under the transport).

⁵ **EMEP:** European Monitoring and Evaluation Programme at the Economic Commission for Europe of the United Nations.

(b) It is not too big⁶ (and, thus, a model containing the species involved in this set of chemical reactions can successfully be treated numerically).

Unfortunately, the question:

**Is any of the chemical schemes proposed until now
such an optimal set or not?**

is still open.

The condensed CBM IV chemical scheme⁷, which is used in some of the models that will be discussed in the following chapters, seems to be feasible: it is not too big and it still contains many of the important species. However, it is probably not the optimal chemical scheme that can be used in a model for studying all important air pollutants. The condensed CBM IV chemical scheme has been proposed by Gery et al. [204]; see also Whitten and Gery [479]. Some of the chemical schemes discussed in Derwent [101,102] may be more appropriate for some large air pollution models.

If one is interested in some special (and small) group of air pollutants (as, for example the long-range transport of sulphur pollutants only; such studies were very popular in the 70'ies and in the beginning of the 80'ies), then it **may** be possible to design, in a rather simple way, optimal (for these special groups of air pollutants) chemical schemes. Let us assume that only the group of pollutants that has been mentioned above (the sulphur pollutants) is to be studied. Then a model that consists of only two species: sulphur dioxide (the primary pollutant) and sulphate (the secondary pollutant) can successfully be used (and has been used in many air pollution models). Moreover, one can assume that it is sufficient to consider only one chemical reaction in this case (the transformation of sulphur dioxide to sulphate). One can furthermore assume that this reaction is **linear** (or, in other words, that it is described mathematically in the model by a term that is proportional to the concentration of sulphur dioxide).

The above example shows that the choice of a chemical scheme is rather simple in some special cases. Unfortunately, very often the opposite is true: the choice of a chemical scheme is a very difficult problem. This problem, the great difficulties that are connected with the choice of a suitable chemical scheme as well as with the treatment of the chemical reactions in large mathematical air pollution models, will also be discussed in the following chapters.

⁶ In fact "big" should be related to the computer available. A set of chemical reactions that is big for a PC or work-station will be quite acceptable if a modern high-speed computer is used.

⁷CBM: Carbon-Bond Mechanism

1.6. Deposition of air pollutants

"Nevertheless the terms 'wet' and 'dry deposition' encompass wider concepts and distinguish the two main pathways by which atmospheric pollutants are returned to the surface of the Earth".

Alan Wellburn: *"Air Pollution and Acid Rain"*. Longman Scientific and Technical, Harlow, Essex, England, 1988, pp. 1-2.

One of the physical processes that takes place during the transport of air pollutants is the deposition. There are two major types of deposition on the surface: (a) wet deposition and (b) dry deposition.

The wet deposition by precipitation, which is also known as precipitation scavenging, is one of the most effective atmosphere cleansing mechanisms. However, while the atmosphere is cleaned, the soil may be acidified by some pollutants and this may be very harmful for some sensitive areas.

The dry deposition of air pollutants is a process that takes place throughout the long-range transport of air pollutants (not only in precipitation periods; i.e. not only when it rains, snows, etc.). There are different mechanisms that drive the dry deposition of air pollutants on the surface. For large particles, for example, removal processes due to gravitational sedimentation are an effective removal source. Impaction on vegetation is another effective process; especially near the ground. Absorption or reaction processes on Earth's surface is a third type of a deposition mechanism. Uptake on Earth's surface depends on many factors as, for example, on:

(i) certain meteorological parameters,

(ii) the physical, chemical and physiological properties of the Earth's surface,

and

(iii) the specific properties of the pollutants under consideration.

More details about the deposition processes as well as descriptions of different mechanisms that can be used to model these processes can be found in [237], [309], [342]-[346], [476] and [477].

1.7. Effects on plants, animals and humans

"Elevated ozone in rural areas is believed responsible for crop losses of 1 to 2 billion dollars annually ".

S. Sillman, J. A. Logan and S. C. Wofsy: *"A Regional Scale Model for Ozone in the United States with Subgrid Representation of Urban and Power Plant Plumes"*. Journal of Geophysical Research, **95**(1990), p. 5731.

The damaging effects of the air pollutants on plants can be divided into two groups:

(i) direct effects

and

(ii) indirect effects.

Certain air pollutants can damage the plants directly. However, the deposition of air pollutants on the ground can cause acidification of the soil, which can damage the plants. These indirect effects are often much more dangerous than the direct effects.

The effects of air pollutants on animals and humans are also either direct or indirect. Again the indirect effects are often much more dangerous than the direct effects. One of the most dangerous effect (in the group of the indirect effects) is the acidification of the fresh water sources.

There is no need to go into more details, because for the purposes of this book it is quite sufficient to emphasize the fact that air pollutants can cause damages to plants, animals and humans when these exceed certain critical levels. The readers that are interested in this topic should study, for example, Becker et al. [25] or Wellburn [474].

The fact that high concentrations and/or depositions of certain air pollutants might cause damage to plants, animals and humans explains why it is necessary to determine, in a sufficiently reliable manner, the critical levels of the concentrations and/or the depositions of the dangerous air pollutants and to keep the concentrations and/or the depositions of the dangerous air pollutants under the critical levels. The latter task is even more important.

1.8. Need for reduction of the air pollution

"Society's concern exists now because urbanization and industrialization have brought together large concentrations of people in small areas. Concentrations of pollutants emitted from many of man's activities thus build up to levels sufficient to have adverse effects on plant, animal, and human health. Pollution concentrations on a global scale are reaching the point of potential serious consequences to the planet itself".

Henry C. Perkins: *"Air Pollution"*. McGraw-Hill Kogakusha, Tokyo, 1974, p. 2.

It has already been mentioned that air pollutants may cause damage to plants, animals and humans when their concentrations and/or depositions are over certain acceptable (or critical) levels. Therefore it is necessary to reduce the concentrations and/or the depositions of the air pollution to the acceptable (or critical) levels (and, moreover, to keep them under the acceptable levels). The fact that the air pollution must be reduced is now well understood. There are several international conventions where the main purpose is to start up and/or to carry on some efforts toward reducing the concentrations and/or the depositions of some of the dangerous air pollutants to the acceptable (or critical) levels. The countries in Western Europe and North America are very active in this field. Of course, this is not a great surprise since all these countries are also highly polluted.

1.9. Reduction of the air pollution versus its economical cost

"Model analysis of atmospheric processes can play an important role in understanding the effect of different sources on regional-scale air quality".

G. R. Carmichael, L. K. Peters and T. Kitada: *"A second generation model for regional-scale transport/chemistry deposition"*. *Atmospheric Environment*, **20**(1986), p. 185.

The reduction of the concentrations of air pollutants is normally an expensive action. Therefore one should analyze carefully all economical consequences of the planned reduction. A careful work has to be done to determine precisely the acceptable (or critical) levels of concentrations of all dangerous air pollutants. Then the concentrations must be reduced to these levels but no more (because of the high cost). Moreover, the reduction to the acceptable (or critical) levels must be optimized: the price of achieving this reduction should be as low as possible. The task of keeping the air pollution under the acceptable (or critical) levels must also be optimized; again, here it is highly desirable to perform this task at the lowest cost possible.

Neither the task of reducing the concentrations to acceptable (or critical) levels in an optimal way nor the task of keeping the air pollution under the acceptable (or critical) levels in an optimal way is an easy task. This is true even when the acceptable (or critical) levels are precisely determined. **These tasks are especially difficult for small countries.** The following example could be given to illustrate this statement.

One of the simple models developed at the National Environmental Research Institute (this model is described in Zlatev [515,517,520]) was used to estimate the contributions of emission sources in different European countries to the total sulphur deposition in different parts of Denmark. Some of the results are shown on Table 1.4. The results are for 1989, but the same experiment was carried out also with meteorological data for 1979 and 1985. The same conclusions (as those that are listed below) can be drawn by using the results for the other two years.

Part of Denmark	All European sources	Danish sources	Polish sources	Sources from Eastern Europe
Funnen	1900	570(30%)	70(4%)	600(32%)
North Zealand	1700	743(44%)	80(5%)	600(35%)
South Jutland	1500	534(36%)	40(3%)	400(27%)
North Jutland	1300	337(26%)	50(4%)	400(31%)

Table 1.4

Total (dry + wet) sulphur deposition (for 1989) in some Danish regions.

The contributions from the sources in East Germany are added to the contributions from the countries in Eastern Europe.

The depositions are measured in mg sulphur per m² per year.

(a) The results that are given in Table 1.4 are total sulphur depositions for the whole 1989. For separate months the results can vary in a quite wide range. For example, the annual contributions (for 1989) of the Polish sulphur emission sources to the total sulphur depositions in the different parts of Denmark vary between 3% and 5%. However, the corresponding contributions are between 8% and 15% when only July 1989 is taken into account. Many factors play an important role when short periods of time are considered (the wind directions probably being the most important).

(b) From the calculated results that are shown in Table 1.4 it becomes quite clear that in a small country (such as Denmark) it is rather difficult, and perhaps impossible, to reduce the pollution to some prescribed levels without cooperative actions together with the other European countries (especially the big countries), because only about one-third (and more precisely some quantities between 26% and 44%) of the total sulphur deposition in Denmark is caused by the Danish sulphur emission sources. By applying some rather crude calculations, it can be concluded that a reduction of the Danish sulphur emission by 30% will lead to a reduction of the total sulphur deposition in the Northern Jutland only by about 8%. Thus, a reduction of the total sulphur deposition in Denmark by 30% will require much greater amount of reduction of the Danish sulphur emission sources if Denmark has to reduce its sulphur deposition alone. It is obvious that this will be an impossible task when the economical cost of the reduction is taken into account. However, while it is quite unrealistic for Denmark alone to reduce its sulphur deposition by about 30%, the task becomes less

difficult (at least not prohibitive) if all countries in the region decide to reduce the sulphur emission sources together.

(c) The results indicate also that the countries in Eastern Europe contribute (to the total sulphur deposition in Denmark) nearly as much as the countries in the Western Europe. Roughly speaking, one-third of the total sulphur deposition in Denmark is caused by the Danish emission sources, one-third can be assigned to emission sources in the countries in Eastern Europe and the rest is emitted from sources in Western Europe.

(d) Finally, it is seen that the sulphur emission sources in a separate country contribute a little to the total sulphur deposition in Denmark (this is true even for such a polluted country as Poland; the contribution from the Polish sources to the different parts of Denmark is only about 3% - 5% on an annual basis).

It should be emphasized that, although common action of several countries makes it possible (in principle, at least) to reduce the air pollution to the critical (or acceptable) levels, the problem of finding an optimal solution of this task remains unresolved. Reliable mathematical models can play a very important role in the efforts to reduce the air pollution to acceptable levels at a low economical cost. This explains why large mathematical models are needed and why many researchers are working on the development of large mathematical models for studying the long-range transport of air pollutants in Europe. It must be emphasized, however, that many more efforts are required in order to get more reliable and more robust air pollution models.

Some other issues concerning the critical (or acceptable) levels of concentrations and/or depositions of damaging air pollutants as well as the use of reliable and robust mathematical air pollution models will be discussed in detail in the next section.

1.10. Critical levels, control strategies and long-range transport models

"The recent years have seen an increasing interest concerning the transport of air pollutants over long distances and the modelling of this phenomenon".

A. Eliassen: *"The OECD Study of Long Range Transport of Air Pollutants: Long Range Transport Modelling"*. Atmospheric Environment, 12(1978), p. 479.

The short discussion in the previous two sections shows clearly (and this has already been emphasized several times) that it is necessary to determine by some sufficiently accurate and robust method certain critical (or acceptable) levels of the concentrations and/or the depositions of the dangerous air pollutants. However, the concept of critical (or acceptable) levels has not been defined precisely in the previous sections (it has been assumed that it is intuitively clear what a critical or acceptable level for the concentrations or the depositions

of a given air pollutant means). It is worthwhile now to try to define this concept in a more rigorous manner. The following definition, which is often used and which describes in a short but adequate way why such a concept is important, can be given.

Definition 1.1

The acceptable (or critical) level for the concentrations or the depositions of a given air pollutant is the *highest* concentration or deposition that will not cause damaging effects to biological systems during a sufficiently long period of time (say, 50 years).

It is very important to emphasize the fact that the critical (or acceptable) level for a given air pollutant is the **highest** concentration or the **highest** deposition of this air pollutant that can be allowed for a certain area in the attempts to avoid all damaging effects.

This requirement (the critical, or acceptable, level is the **highest** concentration or deposition of the air pollutant under consideration that will not cause damages) is very stringent. However, by imposing such a requirement it is ensured that **the air pollution is reduced as much as needed, but no more** (which is necessary because the reduction of the air pollution is in general a very expensive process; see the previous section).

On the other hand (and this is also very important), it is very difficult to determine precisely the critical levels of the concentrations and depositions of some of the air pollutants. In fact, the truth is that it is probably impossible to find the exact critical levels for most of the air pollutants. However, the task of finding good **approximations** of the exact critical levels can be solved (although even this task is extremely difficult; at least for some of the damaging air pollutants).

The determination of the critical levels (or, at least, of some sufficiently accurate approximations of the critical levels) is absolutely necessary. This is, as stated above, also a very difficult task. Unfortunately, it is by far insufficient to determine the critical levels (or even some good and reliable approximations to the exact values of the critical levels). It is also necessary to develop and use in a regular way different control strategies. There are two important cases where it is necessary to apply some reliable and robust control strategies⁸:

Case A	
1.	Control strategies must be used to reduce the depositions of the damaging air pollutants to the critical levels.
2.	If the depositions of the damaging air pollutants have already been reduced to the critical levels, then control strategies must be used to keep them under the critical levels.

⁸ It should be mentioned here that there are many other cases where it is also worthwhile to apply modern control strategies

and

Case B	
1.	Control strategies must be used to reduce the concentrations of the damaging photooxidant pollutants⁹ during episodes to the critical levels.
2.	If the concentrations of the damaging photooxidant pollutants during episodes have already been reduced to the critical levels, then control strategies must be used to keep them under the critical levels.

The task of developing reliable and robust control strategies is very difficult. The fact that there are many interconnections and, thus, the problem is very complicated as illustrated on Fig. 1.1 (more details can be found in Derwent et al. [104]). It is seen from Fig. 1.1 that there are a lot of interrelations between sources, pollutants, effects and receptors.

However, there are at least three other major difficulties that also have to be overcome in connection with the construction of a reliable and robust mathematical model that could successfully be used in different control strategies :

(A) The concentrations and the depositions depend non-linearly on the corresponding emissions (this means that a reduction of one type of emissions by a factor of two will not necessarily lead to the reduction of the involved concentrations and depositions by a factor of two).

(B) The reduction of only one type of emissions will not necessarily solve all related ecological problems (for example, the reduction of the nitrogen emissions can lead to an increase of the ozone concentrations in some parts of the space domain).

(C) Common actions of many countries in a given region are necessary in the efforts to efficiently reduce the air pollution to the critical levels. One country alone, especially a small country (say, Denmark), will not be able to control the critical levels in the long run, because of the transport of air pollutants over its boundaries.

The consequence from (A) is that simple models that are based on an assumption that the concentrations and the depositions of the air pollutants studied depend linearly on the appropriate emissions are not very reliable and probably should not be used in **all** modern control strategies. However, such simple models can be used in connection with studies concerning **some special** air pollutants as, for example, in the case where only sulphur pollutants are of interest. Many such simple models were developed during the end of 70'ies

⁹ A photooxidant pollutant is a pollutant that participates in the initiating step of a photochemical reaction (see also Chapter 2).

and in the beginning of the 80's. The model that is called RAINS (see, for example, Alcamo et al. [4,5]) is a typical representative of this type of models.

The consequence of **(B)** is that a mathematical air pollution model, which is to be used in control strategies, must in general contain all important air pollutants (including here sulphur pollutants, nitrogen pollutants, ammonia-ammonium, ozone and hydro-carbons). It should be stressed that the simple air pollution models that are based on an assumption that the concentrations and/or the depositions depend linearly on the appropriate emissions are oversimplifying the problem also in regard to the number of pollutants that are to be included in the study. An attempt to study the relationship between emissions and concentrations (or depositions) separately for each pollutant is made in these models. For example, see again Alcamo et al. [4,5], one tries to investigate how the reduction of nitrogen emissions will influence the nitrogen concentrations and/or depositions (without taking into account that the variations of the nitrogen emissions will lead to variations of some other concentrations too; the ozone concentrations will certainly be influenced because the ozone participates in many important chemical reactions in which also nitrogen pollutants are involved). In this way the interrelations demonstrated in Fig. 1.1 (see also Derwent et al. [104]) are practically neglected when simple mathematical models are used.

The two consequences that have been drawn above are rather general. These are valid for all kinds of models that are to be used in the control strategies. The next three consequences are related to mathematical models for studying long-range transport of air pollutants.

The consequence of **(C)** is that advanced mathematical models for studying long-range transport of air pollutants (LRTAP) must be used in the control strategies (but note that, while this is necessary, the use of LRTAP models will not be enough in many cases; some other tools, for example short-scale and meso-scale models, will also be needed in the control strategies).

The consequence of **(B)** and **(C)** taken together is that a reliable LRTAP model must contain all important air pollutants (because of the many interconnections between air pollutants and effects; the various interconnections are illustrated in Fig. 1.1 (and fully discussed in Derwent et al. [104]; it should be mentioned here that Fig. 1.1 is taken from Derwent et al. [104]).

The consequence of **(A)**, **(B)** and **(C)** taken together is that a sufficiently large set of chemical reactions, a set which contains also many photooxidant reactions, should be incorporated in a reliable LRTAP model that is to be used in control strategies (this is a very strong requirement; the chemical reactions introduce a high non-linearity in the models and make its numerical treatment very difficult).

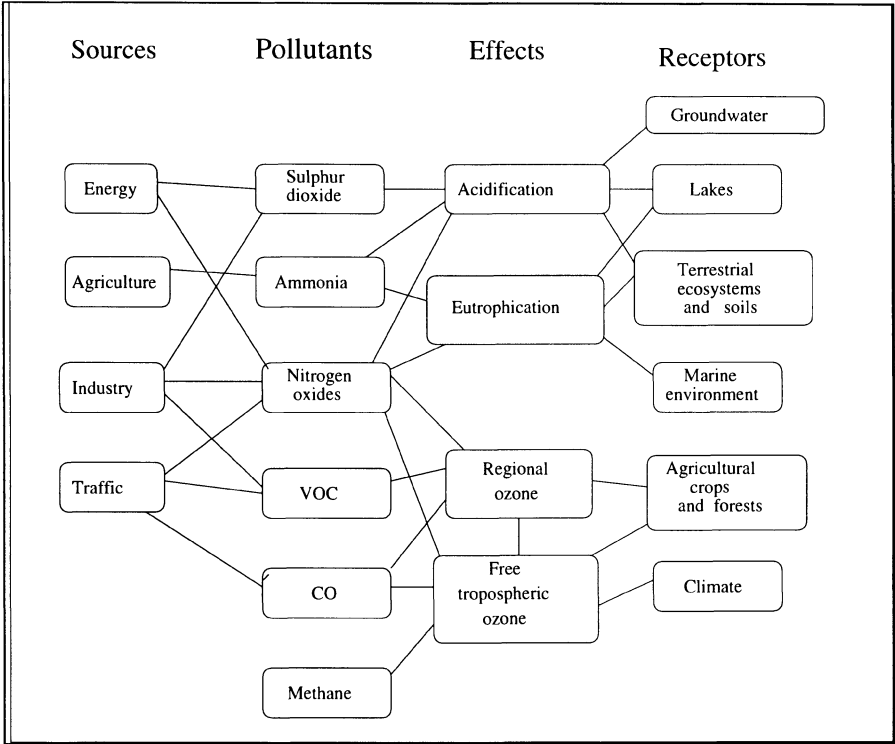


Figure 1.1
Interconnections between sources,
pollutants, effects and receptors
(see also Derwent et al. [104]).

There are many difficulties that are to be overcome when an LRTAP model that satisfies the above conditions is to be developed. **The time requirement** (the CPU time that is needed to run the model on the computer that is available) was the major limiting factor in the development of large LRTAP models until the middle of the 80's. It was practically impossible to use LRTAP models in control strategies before the middle of the 80's (in such strategies one must carry out many runs in a long simulation process, and the computers were not fast enough to accomplish such a task). It explains why simple models, like RAINS (see again Alcamo et al. [4,5]), were developed in this period, and were very popular (such models are still very popular, but it is commonly accepted now that one should be careful when conclusions, which are only based on the use of such models, are drawn).

In the last decade of the 20'th century, the computer time requirement (i.e. the CPU time needed to run a big air pollution problem on the computer that is available) is, at least, not a prohibitive factor any more. The modellers are able to run very big jobs on modern

high-speed computers, and the computers become faster and faster. Computers that can achieve a speed of computations of several **GFLOPS**¹⁰ are already available. There are ambitious plans to develop a new generation of computers that can achieve speeds of computations greater than one **TFLOPS**¹¹. Therefore there is a need for more powerful and more realistic control strategies, **second generation control strategies** (see also Derwent et al. [104]), which are based on the use of advanced and reliable mathematical models and which possess all properties that have been discussed above. Furthermore, and this is even more important, it will be possible to apply the models used in the new control strategies again and again, in many runs and simulations, on modern high-speed computers if the programming work is carefully done (and the great potential power of the new vector and/or parallel computers is correctly exploited to achieve a high speed of computations).

1.11. Concluding remarks and references concerning the air pollution problem

"Odin¹² showed that acid precipitation was a large scale regional phenomenon in much of Europe with well defined source and sink regions, that precipitation and surface waters were becoming acidic, that long-distance (100-2000 km) transport of sulfur- and nitrogen-containing species was taking place over Europe, and that there were marked seasonal trends in deposition of major ions and acidity".

John H. Seinfeld: *"Atmospheric Chemistry and Physics of Air Pollution"*, p. 696. Wiley, New York, 1986.

The air concentrations of many of the pollutants were in general steadily growing during the last decades. Some basic information about the nature of the air pollution problem, i.e. about the emission sources, the transport due to the wind and the transformations (deposition, diffusion and chemical reactions) during the transport, is given in **Section 1.1 - Section 1.6**. Only the information that will be used in the following chapters of this book is presented in the first six sections of Chapter 1. Some further details about the physics of the main processes that are related to the air pollution problems as well as some information about the major chemical reactions that take place during the long-range transport of pollutants in the atmosphere can be found in many other books; as, for example, Pasquill and Smith [347], Seinfeld [391] or Zannetti [503] (see also the references in these books).

High concentrations and/or high depositions of many of the air pollutants can cause damage (first and foremost to plants, animals and humans). The dangerous effects that can

¹⁰ **GFLOPS:** 10⁹ floating point arithmetic operations (additions or multiplications) per second.

¹¹ **TFLOPS:** 10¹² floating point arithmetic operations (additions or multiplications) per second.

¹² **Svante Odin** is a Swedish scientist; one of the pioneers in the systematic studies concerning long-range transport phenomena. Odin established in 1961 a Scandinavian network to measure surface water chemistry (see again Seinfeld, [391]).

be caused (either directly or indirectly) by the air pollutants are beyond the scope of this book. Nevertheless, some of these effects are briefly described in **Section 1.7**. These issues are discussed in detail in many other books treating air pollution phenomena; see, for example, Becker et al. [25] or Wellburn [474].

The fact that different damages can be caused by high concentrations and/or depositions explains why it may be desirable, and in most of the cases it is even necessary, to reduce the concentrations and/or the depositions of certain air pollutants to some acceptable (or critical) levels. This topic is shortly discussed in **Section 1.8**.

However, the concentrations of air pollutants can be reduced to the acceptable levels only at some economical price. The great problems connected with the reduction of the concentrations to acceptable levels are explained in **Section 1.9**. The main conclusion is that it is important to find **an optimal solution** of the problem, i.e.

to reduce the concentrations to the acceptable levels at as a low price as possible.

Such an optimal solution can be achieved by common actions of all countries in the area where the pollution has to be reduced. It has been shown in **Section 1.9** that for the small European countries a common action is a necessary condition. This is not a sufficient condition. Much more efforts are required in order to find an optimal solution. Reliable mathematical air pollution models are to be used in the attempts to get a solution which is close to the optimal one. Mathematical air pollution models and their numerical treatment are the main topic of this book, and these issues will be studied in detail in the following chapters.

Some of the major difficulties that are connected with the reduction of the damaging air pollutants to certain acceptable (or critical) levels are further discussed in **Section 1.10**. It is explained in **Section 1.10** (by using some results that are reported in Derwent et al. [104]) that the major difficulties are caused by the fact that:

(i) there are a lot of interconnections between sources, air pollutants, effects and receptors,

and

(ii) there could be non-linear relations between certain emissions and the corresponding concentrations and/or depositions.

Therefore new and modern control strategies, which are based on the use of advanced mathematical models, are needed in the efforts to prevent complex and irreversible damaging effects caused by simultaneous actions of several air pollutants.

CHAPTER 2

MATHEMATICAL MODELS FOR STUDYING THE LONG-RANGE TRANSPORT OF AIR POLLUTANTS

"It is not unusual to hear qualified scientists making statements such as 'Why do we need to model that? Let's measure it; that's all we need,' 'Models do not work,' etc. These statements imply unscientific thinking. Science involves development of theory (or 'models') based on (1) the empirical interpretation of experimental data, (2) the generalization of experimental relationships or (3) pure speculative thinking subsequently confirmed by experimental results. The advancement of science is not the consequence of monitoring activities, even though the collection of good reliable experimental data is often (but not always) a necessary (but not sufficient) condition to it."

Paolo Zannetti: *"Air Pollution Modeling: Theories, Computational Methods and Available Software"*, Computational Mechanics Publications, Southampton-Boston, 1990.

The long-range transport of air pollutants, the LRTAP, is a very complicated physical phenomenon. Some air pollutants can be transported over long distances to remote areas that are far away from the big emission sources. Moreover, under certain circumstances the transported air pollutants may lead (directly or indirectly) to some damaging effects on plants, animals and humans. Therefore, it is important to be able to study the LRTAP carefully. Mathematical air pollution models are indispensable tools in the investigation of long-range transport phenomena. The analytical description of some types of such large models by systems of partial differential equations will be discussed in this chapter.

It has already been mentioned (in Chapter 1) that the long-range transport of air pollutants, the LRTAP, can be considered as a result of the interaction of five physical processes:

- (1) advection,
- (2) diffusion,
- (3) deposition,
- (4) emission

and

- (5) chemistry.

The first task that has to be solved during the development of a long-range transport model is to describe mathematically these five physical processes. The principles (or, at least, some of the principles) that can be applied in connection with the mathematical description of each of these five physical processes will be discussed in the first eight sections of this chapter. Then a combined mathematical model for studying long-range transport phenomena

in connection with air pollutants will be introduced in Section 2.9 and briefly discussed there.

It will be shown that the combined mathematical model for studying the long-range transport of air pollution is normally described, like many other important physical processes, by a system of partial differential equations (a system of PDE's). The number of equations in this system is equal to the number of species that are studied by the model. Thus, the system becomes bigger when one wants to study more species in the model. Even when the number of species that are to be studied is not very large, the numerical treatment of a model for studying the long-range transport of air pollution is a difficult task, because the discretization of the system of partial differential equations leads to huge systems of ordinary differential equations that are to be integrated numerically over a long time-intervals. The numerical treatment of large long-range transport models will be discussed in the following chapters of this book.

2.1. Long-range transport and reduction of air pollution

The physical phenomenon which is well-known under the name "long-range transport of air pollution (LRTAP)" can be divided, as mentioned in Section 1.1, into three major stages. These three major stages are briefly described in Table 2.1.

It is seen from Table 2.1 that five major physical processes (emission, advection, diffusion, deposition and chemical reactions) are forming the LRTAP; this was also mentioned in Chapter 1. The first task that must be solved during the development of an air pollution model is to describe all these processes mathematically.

Stage 1	Emission
Stage 2	Transport due to the wind (advection)
Stage 3	Transformations during the transport:
	3.1. Diffusion
	3.2. Deposition
	3.3. Chemical reactions

Table 2.1
The three major stages during
the long-range transport of air pollutants.

The LRTAP will, in the following sections of this chapter, be described mathematically by a system of partial differential equations (PDE's): the number of equations in the system of PDE's is equal to the number of species that are to be studied by the model. Furthermore, the system of PDE's will be defined in a large space domain. Indeed, only when the space domain, in which the system of PDE's is defined, is sufficiently large, one can study long-range transport phenomena.

Certain mathematical functions are to be used in the description of the five physical processes in a large LRTAP model. It is normally assumed that these functions (as, for example, the wind velocities) are known in the whole space domain and in the whole time-interval. Such a requirement (certain mathematical functions to be known in the whole space domain and in the whole time-interval) is usually very stringent when practical problems are to be treated. This can be demonstrated by the example given below.

Let us assume that the LRTAP model is two-dimensional. This means that the long-range transport of air pollutants is studied on a two-dimensional space domain. Furthermore, let us assume that the space domain is discretized by an equidistant grid consisting of equal grid-squares. As an illustration the actual grid that has been used in **the Danish Eulerian Model** (by which the long-range transport of air pollutants over Europe has been studied in Harrison et al. [228], Zlatev and Christensen [537], Zlatev et al. [538,539]) is given in Fig. 2.1. It should be mentioned here that the grid shown in Fig. 2.1 is a 32x32 grid. A finer spatial grid, a 96x96 grid, is used now in the discretization of **the Danish Eulerian Model**.

Consider the horizontal advection process, i.e. the transport of pollutants in the atmosphere that is directly caused by the horizontal components of the wind. The horizontal wind velocities $u(x,y,t)$ and $v(x,y,t)$ (i.e. the wind velocities along the Ox and the Oy axes respectively) are to be used in order to describe the advection mathematically. Both $u(x,y,t)$ and $v(x,y,t)$ are to be considered (in the model) as known mathematical functions that can be calculated at every point (x,y) of the space domain and at every time t . This example shows clearly that the requirement that both $u(x,y,t)$ and $v(x,y,t)$ are known functions on the whole space domain and on the whole time-interval is a rather stringent requirement and as a rule it cannot be satisfied in practice. This means that one should try to relax this requirement. Of course, such a situation arises not only when the horizontal advection process is to be handled mathematically, but also when the other physical processes involved in the study of long-range transport of air pollution are to be treated in a mathematical model.

Having some discretization of the space domain and also some discretization of the time-interval, one can relax the requirement that $u(x,y,t)$ and $v(x,y,t)$ are known on the whole space domain and the whole time-interval by a requirement that sufficiently accurate approximations to the values of these functions can be calculated at all grid-points (x_i, y_j, t_k) where

$i=1,2,\dots,I$ (I being the number of grid-points on the grid-lines parallel to the Ox axis),

$j=1,2,\dots,J$ (J being the number of grid-points on the grid-lines parallel to the Oy axis),

$k=1,2,\dots,K$ (K being the number of grid-points along the time axis).

It is clear that the problem is substantially simplified when such an assumption is made, because the requirement to know analytical expressions for the functions $u(x,y,t)$ and $v(x,y,t)$ which are valid in the whole space domain and in the whole time-interval is replaced by a requirement to know sufficiently accurate approximations to the exact values of these functions on **finite** number of points.

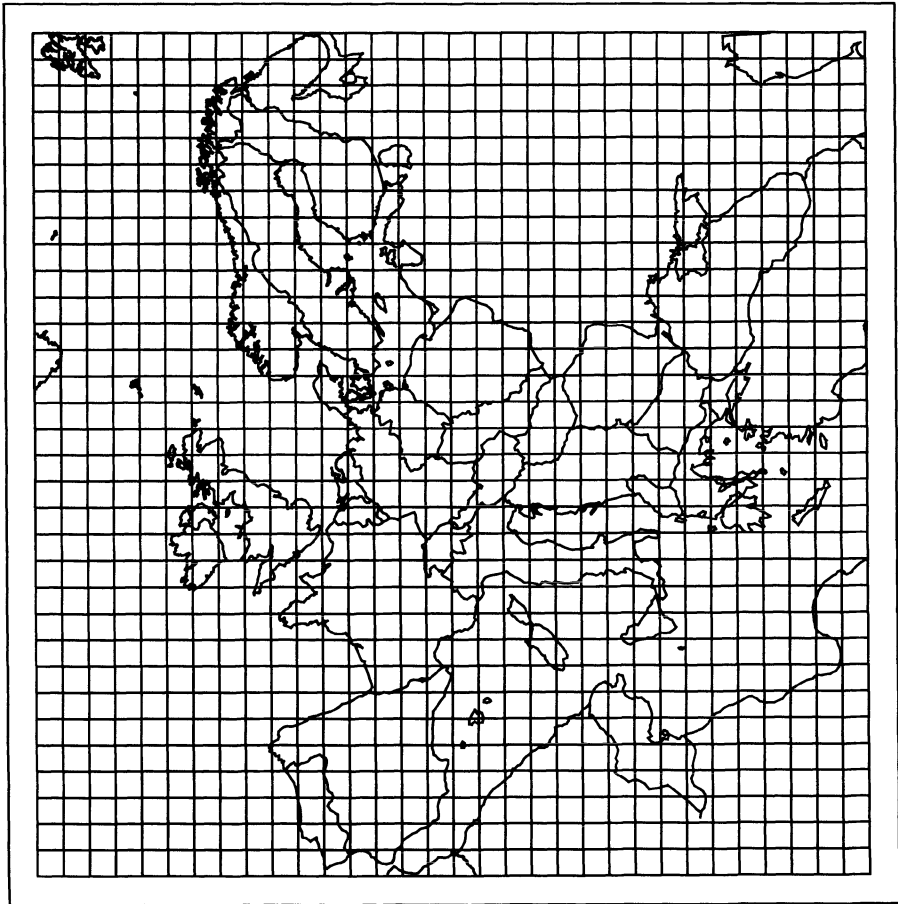


Figure 2.1.
The space domain that is used in the Danish Eulerian Model
discretized by an equidistant (32x32) grid;
in the latest version of the Danish Eulerian Model
an equidistant (96x96) grid is applied.

The two dimensional case has been discussed above. Quite similar ideas can also be applied when three-dimensional models are to be treated numerically.

It should be stressed that the requirement for sufficiently accurate approximations is important because:

(a) if the approximations are not sufficiently accurate, then the results obtained by the mathematical model will normally be unreliable

and

(b) if the approximations are much more accurate than the accuracy required for the model output, then perhaps the high accuracy of the values of $u(x_i, y_j, t_k)$ and $v(x_i, y_j, t_k)$ is obtained by using some additional measurement stations and/or some extra computational efforts that could be saved (it must be emphasized here that reducing the computational work as much as possible is one of the most important tasks during the numerical treatment of large models for studying long-range transport of air pollutants; this task will be discussed in detail in the following chapters).

In the same way, as for the known functions that are used during the mathematical treatment of the advection process, the requirement to the known functions connected with the other physical processes can be relaxed; i.e. sufficiently accurate approximations to these functions on the grid-points can also be used when the deposition, diffusion, emission and chemical reactions are introduced in the model.

Reliable mathematical models for studying the long-range transport of air pollutants in a prescribed space domain (say, the long-range transport in Europe) can be used for many different purposes. However, the most important task (see also Chapter 1) is

the task of reducing the air pollution in a prescribed area to some acceptable levels at a minimal economical price.

This task is so important that there is nearly no need to justify further (by describing into detail the other tasks that can be treated by using large air pollution models) the usefulness of developing and applying reliable mathematical models. Nevertheless, some other tasks are listed (together with the above task) in Table 2.2; this list can be continued.

Before starting to discuss the mathematical models for studying air pollution phenomena, it is necessary to explain why both mathematical models and measurements are to be used when long-range transport of air pollutants is studied.

2.2. Use of models versus use of measurements

Some information about the distribution of the concentrations of different air pollutants in a given area can be obtained by studying measurements taken in different parts of the area. This is true both in connection with the determination of air pollution levels of the concentrations in the area and even in connection with the determination of the amount of the transported (by long-range transport) air pollution.

A necessary condition for carrying out successfully such studies is to have a sufficiently large network of measurement stations in the area under consideration. However, even when a large network of measurement stations is available, there are still some tasks that cannot be studied by using measurements only (or, at least, cannot be studied in an easy way only by using measurements). In fact, some of the tasks that are listed in Table 2.2 can be studied efficiently only by using reliable mathematical models. This is especially true for the important Task 5.

Task	Short description of the task
1	Obtaining information about the levels of the concentrations of different air pollutants in different parts of the space domain.
2	Studying the diurnal and seasonal variations of the concentrations of certain air pollutants.
3	Calculating the contributions of the emission sources in one part of the space domain to the concentrations and depositions in the surrounding parts.
4	Studying the dependence of the concentrations of certain air pollutants (say, ozone) on different physical parameters (for example the temperature).
5	Finding a way to reduce the air pollution in a prescribed area to some acceptable level at a minimal economical price.

Table 2.2
Some of the important tasks that can be studied
by using reliable mathematical models.

In the case where one is interested in studying what will be the result of reducing (by a prescribed amount) the emissions from all sources in a given area on the concentrations in the same area (or, what is even more important, on the concentrations in the surrounding areas), reliable mathematical models are the most convenient tool for achieving the required information. Note that, as a rule it is necessary to have the answers to the above questions

for many different reductions of emission sources. In some situations also the effect of several different increases of emission sources in a given area should be studied. The latter situation occurs when the future development of an industry in a given area is to be planned.

The above discussion (together with the examples given above) shows clearly that the application of reliable mathematical models is more preferable than the use of a large network of measurement stations; in many situations, at least. It should also be emphasized here that the use of a mathematical model is as a rule much cheaper than the use of observations from a large network of measurement stations. However, the requirement for **reliability** of the mathematical model is essential. Therefore the question:

How can the reliability of the concentrations and depositions that are calculated by a given mathematical model be checked?

is important and has to be answered.

Checking the reliability of the concentrations calculated by a given mathematical model can be performed in a natural way by comparing the concentrations calculated by the model with measurements. If there are sufficiently many measurement stations in the space domain covered by the model and if there are good reasons to believe that the measurements are reliable, then checks based on comparisons of calculated concentrations and measurements will give a good information on whether the mathematical model is reliable or not.

If the comparisons of calculated concentrations with measurements indicate that the mathematical model is not reliable, then an investigation of the model must be made. One should try once again to check everything in the mathematical model:

- (i) the physical parameters,
- (ii) the mathematical description,
- (iii) the numerical procedures

and

- (iv) the programming.

If the comparisons with measurements indicate that the mathematical model is reliable, then one may proceed by setting up new experiments for which there are **no measurements** (as, for example, experiments with different emission sources by which one can study how the concentrations and depositions of various air pollutants will change when the emission sources are reduced or increased). This important task cannot be handled (or, at least, it is difficult to be handled) without a reliable mathematical model.

The main conclusion is that, while very valuable information about the air pollution in a given area can be obtained by using measurements, the use of mathematical models is often much more powerful (one can also obtain useful information in cases where measurements do not exist when a mathematical model is used). However, the reliability of the results obtained by a mathematical model must be checked carefully and here the measurements play an important role: they could be used to check the reliability of the concentrations and depositions calculated by a mathematical model.

2.3. Modelling the transport in the atmosphere

The transport of various air pollutants over long distances in the atmosphere (the physical process that is also called advection) is one of the most important physical processes when long-range transport of air pollutants, LRTAP, is studied. In this section the mathematical formulation of the transport phenomena will be discussed.

Let us consider the situation where a mathematical model containing **advection only** is to be constructed. This means that there will be no diffusion, no deposition, no emission sources and no chemical reactions in such a model. Of course, this is an artificial situation. In a real situation all other physical processes are represented and play a considerable role in the global model. However, by such a consideration (of a pure advection process) one can both simplify the explanation of the main features of the advection process and facilitate the construction of the global mathematical model for studying long-range transport of air pollutants. For example, it is obvious that one can assume that the advection part of a mathematical LRTAP model does not depend on the particular air pollutants that are transported: this means that the transport of **any** of the air pollutants studied can be described mathematically by using the same partial differential equation. Thus, although the global mathematical LRTAP model is described by a system of PDE's (the number of the equations in the system being equal to the number of air pollutants involved in the study), the following statement holds.

It is quite sufficient to consider only a single partial differential equation in order to explain everything concerning the advection process.

Let us denote by $c(x,y,z,t)$ the concentration of a given air pollutant at the point $(x,y,z) \in D \subset \mathbf{R}^3$ and at time $t \in [0,T]$, where D is a space domain in the three-dimensional Euclidean space $\mathbf{R}^3$ and $[0,T]$ is a time-interval with some $T > 0$. Let us denote the wind velocities along the three coordinate axes by $u(x,y,z,t)$, $v(x,y,z,t)$ and $w(x,y,z,t)$, respectively. Furthermore, let us also assume that the wind velocities are known functions defined in the whole space domain D and on the whole time-interval $[0,T]$.

If there is no danger of misunderstanding, then the independent variables x , y , z and t will as a rule be omitted; i.e. the abbreviations c , u , v and w will be used instead of $c(x,y,z,t)$, $u(x,y,z,t)$, $v(x,y,z,t)$ and $w(x,y,z,t)$, respectively.

When large LRTAP models are considered, the space domain is very often a parallelepiped in the three dimensional Euclidean space $\mathbf{R}^3$. If an assumption is made that the space domain is a parallelepiped, then one can assume that the space domain D is defined by

$$(2.1) \quad D = \{ (x,y,z) \mid x \in [0,X], \quad y \in [0,Y], \quad z \in [0,Z] \},$$

where $X > 0$, $Y > 0$ and $Z > 0$. It is clear that the assumption that D is defined by (2.1) (or, more precisely, that the left end-points of the space intervals along the coordinate axes are zeros) is not a restriction: if the space domain D is a parallelepiped, then there exists a transformation of the space coordinates such that (2.1) holds.

Under the assumptions made above, the long transport of air pollutants that is due to the wind can adequately be described by the following partial differential equation (PDE) and, as stated above, this partial differential equation can be used in the description of the transport of any air pollutant in the atmosphere:

$$(2.2) \quad \partial c / \partial t = - \partial (uc) / \partial x - \partial (vc) / \partial y - \partial (wc) / \partial z, \\ \text{for } \forall (x,y,z) \in D \subset \mathbf{R}^3 \quad \text{and for } \forall t \in [0,T].$$

The above equation can be simplified (and it is, in fact, very often simplified in the mathematical models; see, for example, Marchuk [307]) by assuming that a conservation law is satisfied for the wind velocities in the lower parts of the atmosphere:

$$(2.3) \quad \partial u / \partial x + \partial v / \partial y + \partial w / \partial z = 0, \\ \text{for } \forall (x,y,z) \in D \subset \mathbf{R}^3 \quad \text{and for } \forall t \in [0,T].$$

By substituting equation (2.3) in the three-dimensional advection model (2.2) the following PDE can be obtained:

$$(2.4) \quad \partial c / \partial t = - u \partial c / \partial x - v \partial c / \partial y - w \partial c / \partial z, \\ \text{for } \forall (x,y,z) \in D \subset \mathbf{R}^3 \quad \text{and for } \forall t \in [0,T].$$

It is clear that (2.4) is simpler than (2.2). The assumption that (2.3) is satisfied, which is necessary to obtain (2.4), is, as mentioned above, often used in the development of air pollution models.

The mathematical model that is defined by (2.2) describes a pure advection process (of any air pollutant) in the three-dimensional Euclidean space. Two-dimensional models can also be considered. In the latter case the functions involved in the model are $c(x,y,t)$, $u(x,y,t)$ and $v(x,y,t)$. These functions are considered now at any point $(x,y) \in D \subset \mathbf{R}^2$ and at any time $t \in [0,T]$, where D is a space domain in the two-dimensional Euclidean space $\mathbf{R}^2$ and $[0,T]$ is again a time-interval with some $T > 0$. Also here the abbreviations c , u and v will be used (when there is no danger for misunderstanding) instead of $c(x,y,t)$, $u(x,y,t)$ and $v(x,y,t)$, respectively. Finally, it will normally be assumed that the two-dimensional space domain D is rectangular. The rectangular space domain D can be defined, in a similar way as in the three-dimensional case, by

$$(2.5) \quad D = \{ (x,y) \mid x \in [0,X], \quad y \in [0,Y] \},$$

with $X > 0$ and $Y > 0$. Also in this situation the assumption that D is defined by (2.5) is not a restriction: if the space domain D is a rectangle, then there exists a transformation of the space coordinates such that (2.5) holds.

A two-dimensional model can **formally** be obtained from the three-dimensional model that is represented by (2.2). It is simply necessary to remove the last term in the right-hand side of the partial differential equation (2.2) in order to obtain a two-dimensional model. One should also use the assumptions made above for the functions c , u and v as well as for the space domain D . The result is given by:

$$(2.6) \quad \partial c / \partial t = - \partial (uc) / \partial x - \partial (vc) / \partial y,$$

for $\forall (x,y) \in D \subset \mathbf{R}^2$ and for $\forall t \in [0,T]$.

It is easy to obtain, formally at least, a two-dimensional advection model from the advection model (2.4). As in the previous case, one should again remove the last term in the right-hand side of (2.4). Moreover, one should also use the assumptions made above for the functions c , u and v as well as for the space domain D . Under these assumptions the two-dimensional version of (2.6) can be written as:

$$(2.7) \quad \partial c / \partial t = - u \partial c / \partial x - v \partial c / \partial y,$$

for $\forall (x,y) \in D \subset \mathbf{R}^2$ and for $\forall t \in [0,T]$.

The derivation of (2.7) from (2.4) is, as stated above, a formal derivation only. There are some difficulties when the use of (2.7) is to be justified. The main problem is that while

the assumption made by (2.3) is a plausible one (see again Marchuk [307]), it will **not** be plausible to assume that the relationship:

$$(2.8) \quad \partial u / \partial x + \partial v / \partial y = 0, \\ \text{for } \forall (x, y) \in D \subset \mathbf{R}^2 \quad \text{and for } \forall t \in [0, T],$$

also holds. Therefore one has to use (2.2) and (2.3) together with an assumption that the concentrations are evenly distributed in the vertical direction (and this last assumption can be justified in certain situations) in order to derive a two-dimensional version of (2.4). This means that the following assumption is to be introduced:

$$(2.9) \quad \partial c / \partial z = 0$$

for any fixed point (x, y) in the space domain, $x \in [0, X]$, $y \in [0, Y]$, and for any fixed value of t in the time-interval, $t \in [0, T]$.

After this discussion it is clear that the use of (2.7) can be justified (in some situations at least) by applying (2.2), (2.3) and (2.9) in its derivation. However, it should be stressed that it is probably better to apply (2.6) instead of (2.7) when two-dimensional LRTAP models are used.

In some situations it is useful to consider even one-dimensional advection equations:

$$(2.10) \quad \partial c / \partial t = - \partial (uc) / \partial x, \\ \text{for } \forall x \in [0, X] \quad \text{and for } \forall t \in [0, T],$$

or

$$(2.11) \quad \partial c / \partial t = - u \partial c / \partial x, \\ \text{for } \forall x \in [0, X] \quad \text{and for } \forall t \in [0, T].$$

Note that the space domain, in the case where one-dimensional advection transport is considered, is defined by

$$(2.12) \quad D = \{ x \mid x \in [0, X] \},$$

with some $X > 0$.

Of course, **it is not realistic** to study general phenomena concerning LRTAP by using one-dimensional models. However, one-dimensional models can successfully be used to study some special cases and/or to solve many auxiliary problems. Such models can be used, for example, to study short-scale (or even meso-scale) transport in periods of time when the wind does not change its direction. If the advection equation is to be solved numerically, then the simple equations (2.10) and (2.11) can be used as a first check of the reliability of the numerical method chosen.

In all equations which were derived in this section it was assumed that the wind velocities u , v and w are known functions; i.e. it is assumed that the values of these functions can be calculated for all points of the space domain D and for all values of the time-variable t ($t \in [0, T]$). It has already been pointed out, see Section 2.1, that in fact it is not possible to satisfy such a requirement when large LRTAP models are treated, because the wind velocities should either be measured in some way (and this is done at many meteorological stations) or calculated, on a given grid, by some weather forecasting model. This means that only a finite number of values of the wind velocities is as a rule known when long-range transport of air pollutants is to be studied. Therefore the requirement that the values of the wind velocities are to be calculated exactly for all points (x, y, z) of the space domain D and for all values of the time-variable t ($t \in [0, T]$) is usually replaced by a requirement that sufficiently accurate approximations of these three known functions can somehow be calculated at a finite number of grid-points (see again the discussion in Section 2.1).

The solution of any of the advection equations described in this section is uniquely determined only if an initial value condition is given, which means that the concentrations at the starting point of time-interval ($t = 0$) must be given for all points of the space domain D . Therefore the advection equations given above must be handled together with an initial value condition given by one of the following relations:

$$(2.13) \quad c(x, y, z, 0) = p(x, y, z) \quad (\text{in the three-dimensional case}),$$

$$(2.14) \quad c(x, y, 0) = p(x, y) \quad (\text{in the two-dimensional case}),$$

$$(2.15) \quad c(x, 0) = p(x) \quad (\text{in the one-dimensional case}).$$

It is assumed here that p is a known function (such an abbreviation, where the arguments are omitted, will often be used; from the context it should be clear what case is considered). When large air pollution problems are studied, there will be similar problems with the known function p as those concerning u , v and w ; i.e. in practice it will be difficult to calculate the values of function p at all points of the space domain D . However, the same procedure (as that used in the calculation of the wind velocities) can be used in connection with the initial concentrations p . The requirement that the values of p must be known at all points of D is replaced with a weaker requirement that some

sufficiently accurate approximations to the exact values of p can be calculated by some numerical procedure (or in some other way) at a finite number of grid-points in the space domain D . The following procedure is followed in many large air pollution models in an attempt to get sufficiently accurate initial concentrations: (i) one starts with some rather crude approximations of the initial values, (ii) one runs the model several days (five days have been used in **the Danish Eulerian Model**; see Zlatev et al. [538,539]), (iii) one accepts the concentrations thus found as improved initial values and, after that, one starts the calculations of the concentrations which are actually desired.

The advection equations that are discussed in this section can generally not be solved exactly (or analytically). This is so, because exact (or analytical) solutions can only be obtained if very stringent requirements are imposed on the known functions u , v and w (so stringent that these functions will not, in general, have anything to do with the actual wind velocities when such assumptions are imposed). Therefore numerical methods must be used in order to obtain numerical solutions. At the same time, it must be emphasized that test-problems with known analytical solutions are needed: such test-problems may be very useful when the accuracy of the numerical methods chosen is to be investigated.

Consider the one dimensional advection equation (2.11) and assume that the wind velocity is constant; i.e.

$$(2.16) \quad u(x, t) = u_0 \\ \text{for } \forall x \in [0, X] \quad \text{and for } \forall t \in [0, T],$$

Let us also assume that some initial field of concentrations $p(x)$ is defined by (2.15). Equation (2.11) is written as

$$(2.17) \quad \partial c / \partial t = - u_0 \partial c / \partial x, \\ \text{for } \forall x \in [0, X] \quad \text{and for } \forall t \in [0, T].$$

when the assumption given by (2.16) is made.

The analytical solution of the one-dimensional advection equation which is defined by (2.17) and (2.16) is given by the following function:

$$(2.18) \quad c(x, t) = p(x - u_0 t) \\ \text{for } \forall x \in [0, X] \quad \text{and for } \forall t \in [0, T],$$

The one-dimensional advection equation (2.17) (with a known analytical solution) is often used to verify the applicability of a numerical method in the treatment of the advection part of large air pollution problems or to perform some theoretical investigations; see, for example, Kreiss and Olinger [270]. This equation will be discussed in Chapter 4 and will be used to justify some results in Chapter 5.

Although the one-dimensional test-problem can often provide sufficient information concerning some properties of the numerical methods tested, there are situations when this test is not sufficient (especially when two- and three-dimensional advection equations are studied). A simple test-problem with a known analytical solution is often needed when two-dimensional advection equations are studied. A very convenient two-dimensional test-problem has been proposed simultaneously in 1968 by Crowley [81] and Molenkamp [320]. This test-problem can be described by equation (2.7) as follows. Assume that

$$(2.19) \quad D = \{ (x,y) \mid x \in [0,2], \quad y \in [0,2] \},$$

$$(2.20) \quad T = 2\pi,$$

$$(2.21) \quad u(x,y,t) = y-1, \quad v(x,y,t) = 1-x \\ \text{for } \forall (x,y) \in D \subset \mathbf{R}^2 \quad \text{and for } \forall t \in [0,T].$$

The use of (2.19)-(2.21) in (2.7) results in the following two-dimensional test-problem:

$$(2.22) \quad \partial c / \partial t = - (y-1) \partial c / \partial x - (1-x) \partial c / \partial y, \\ \text{for } \forall x \in [0,2], \quad \forall y \in [0,2] \quad \forall t \in [0,2\pi].$$

Assume that (x,y) is any point in the space domain D which is different from its centre $(1,1)$; i.e. $(x,y) \neq (1,1)$. It can be proved that the concentration in any such point (x,y) will rotate, with a constant angular velocity, along a circle with a centre $(1,1)$ and a radius $r = [(x-1)^2 + (y-1)^2]^{0.5}$. This means that if a given initial concentration field is defined by the function $p(x,y)$, i.e. if $c(x,y,0) = p(x,y)$, then after time $t = 2\pi$ the solution will be $c(x,y,2\pi) = p(x,y)$. Thus, this test problem can also be used (and in fact has often been used) to study the accuracy of different numerical methods for two-dimensional advection equations; see, for example, Zlatev et al. [529].

These two test problems are quite sufficient for the purposes in this chapter: they show that one **can** construct some simple test problems with known analytical solutions to check the reliability of the numerical methods used. More details will be given in the following chapters where the accuracy of the numerical methods for the solution of advection equations will be discussed (see, for example, Chapter 4).

Finally, the problem of finding suitable boundary conditions should be mentioned. No boundary condition is needed on the so-called out-flow parts of the boundary (the parts of the boundary of the space domain where the concentrations are transported outside the space domain by the wind). Some boundary condition must be specified on the in-flow parts of the boundary. However, boundary conditions are in practice needed even on the out-flow parts (or, at least, some special treatment of certain grid-points that are close to the boundaries is necessary) when numerical methods are used. This problem will be studied in more detail in connection with the numerical treatment of the advection equations.

2.4. Modelling the diffusion phenomena

When pollutants are released into the atmosphere from the emission sources inside the space domain studied by the air pollution model, these are, first and foremost, transported by the wind. This is the advection physical process that has been described in the previous section. Three other important physical processes take place during the transport of air pollutants in the atmosphere (see Section 2.1 and Chapter 1). One of these processes is the diffusion (i.e. the dispersion of the air pollutants in the atmosphere).

Let us assume that only diffusion takes place (i.e. there is no advection, no deposition, no emission and no chemistry). Again, as in the previous section, it should be stressed that this is an artificial situation, because diffusion should be considered together with the other four basic physical processes. However, the study of the diffusion only has several advantages:

- (i) one can simplify the explanation of the main features of the diffusion process

and

- (ii) one can facilitate the construction of the global mathematical model for studying long-range transport of air pollutants.

For example, it can very often be assumed (in the same way as in the previous section) that the diffusion part of a mathematical LRTAP model does not depend on the particular air pollutants that are transported: this means that the transport of **any** of the air pollutants studied can be described mathematically by using the same partial differential equation. Thus, although the global mathematical LRTAP model is described by a system of PDE's (the number of the equations in the system being equal to the number of air pollutants involved in the study),

it is quite sufficient to consider only a single partial differential equation in order to explain everything concerning the diffusion process.

It should be reiterated here that precisely the same arguments were used in the previous section in order to justify the use of a single partial differential equation when only pure advection is studied.

Let us denote, as in the previous section, the concentration of a given air pollutant at the point $(x,y,z) \in D \subset \mathbf{R}^3$ at time $t \in [0,T]$ by $c(x,y,z,t)$, where D is a space domain in the three-dimensional Euclidean space $\mathbf{R}^3$ and $[0,T]$ is a time-interval with some $T > 0$. Let us denote the diffusivity coefficients along the three coordinate axes by $K_x(x,y,z,t)$, $K_y(x,y,z,t)$ and $K_z(x,y,z,t)$, respectively. Furthermore, let us also assume that the diffusivity coefficients are known functions defined in the space domain D and on the time-interval $[0,T]$.

If there is no danger for misunderstanding (see also the previous section), then the independent variables x , y , z and t will as a rule be omitted; i.e. the abbreviations c , K_x , K_y and K_z will be used instead of $c(x,y,z,t)$, $K_x(x,y,z,t)$, $K_y(x,y,z,t)$ and $K_z(x,y,z,t)$, respectively.

When large LRTAP models are considered, the space domain is normally a parallelepiped in the three dimensional Euclidean space $\mathbf{R}^3$. If an assumption is made that the space domain is a parallelepiped, then the space domain D can be defined, as in the previous section, by (2.1).

Under the assumptions made above, the diffusion process can be described (see, for example, Pasquill and Smith [347]) by the following partial differential equation (and, as stated above, this equation holds for the diffusion, in the atmosphere, of any pollutant):

$$(2.23) \quad \frac{\partial c}{\partial t} = + \frac{\partial (K_x (\partial c / \partial x))}{\partial x} + \frac{\partial (K_y (\partial c / \partial y))}{\partial y} \\ + \frac{\partial (K_z (\partial c / \partial z))}{\partial z}.$$

It is often assumed that K_x and K_y are non-negative constants. In the models described in Zlatev [517,520], Zlatev and Berkowicz [526] and Zlatev et al. [525,527-535] it is assumed that

$$(2.24) \quad K_x = K_y = 30000 \text{ m}^2/\text{s}.$$

Similar values for the horizontal diffusivity coefficients are often used (or it is recommended to use such values) in large LRTAP models (see, for example, Gifford [205]).

The treatment of K_z is more difficult. Several suggestions are discussed in Pasquill and Smith [347]. These are

$$(2.25) \quad K_z = \text{const.},$$

$$(2.26) \quad K_z = z^\alpha, \quad 0 \leq z \leq Z,$$

$$(2.27) \quad K_z = (Z-z)^\alpha, \quad 0 \leq z \leq Z,$$

$$(2.28) \quad K_z = z(Z-z), \quad 0 \leq z \leq Z,$$

$$(2.29) \quad \begin{aligned} K_z &= z, & 0 \leq z \leq Z/2, \\ K_z &= Z-z, & Z/2 < z \leq Z. \end{aligned}$$

There are many other ways to determine the vertical diffusion. It is not necessary to discuss all of them here. A rather crude, but still applicable, way to introduce a vertical diffusion is the following. Assume that the height of the mixing layer at the point (x,y) and in time t is $H_{\text{mix}} = H_{\text{mix}}(x,y,t)$. Then the vertical diffusivity coefficient $K_z(x,y,z,t)$ can be defined by

$$(2.30) \quad K_z(x,y,z,t) = K \quad \text{if} \quad z \leq H_{\text{mix}}(x,y,t)$$

and

$$(2.31) \quad K_z(x,y,z,t) = 0 \quad \text{if} \quad z > H_{\text{mix}}(x,y,t),$$

where K is a positive-valued function (see, for example, Zlatev et al. [530,531], where $K = 300 \text{ m}^2/\text{s}$ has been used in day-time, while the value used during night-time has been considerably smaller; $K = 50 \text{ m}^2/\text{s}$).

More sophisticated modules for calculating the vertical diffusion coefficients can be found, for example, in Seinfeld [391]. Such modules can be applied in connection with any of the numerical methods used to treat the vertical transport. One must be aware, however, that a more sophisticated module will not necessarily improve the model results. In fact, the results may become worse if a sophisticated module (even a perfect one; from a theoretical

point of view) is just mechanically attached to the model. This is so because the use of a correct physical module in a large air pollution model is only a necessary condition for obtaining better results. The sensitivity of the modules to uncertainties in the input data is another important requirement. Assume, for example, that one quantity A depends on the third power of another quantity B ; i.e. $A=F(B^3)$. Assume also that the quantity B is measured and that the uncertainties in the measurements are 20%. Then the uncertainties of the calculated values of A will be up to a factor of two-three, and one should be very careful when deciding whether the module should be attached to the model (in spite of the fact that the module is physically correct when B is sufficiently accurate). This example shows that it is not easy to decide which module to choose: a module which is potentially very accurate but also very sensitive to the accuracy of the input data or a module which is more robust but not very accurate. The right decision in such situations (which appear many times when a large air pollution model is to be developed; not only in the diffusion part) may have great influence on the global performance of the model.

As in the case where the advection process was studied (see the previous section), it is also possible to consider diffusion phenomena in two dimensions (i.e. horizontal diffusion only) and even in one dimension. The corresponding partial differential equations are

$$(2.32) \quad \partial C / \partial t = + \partial (K_x (\partial C / \partial x)) / \partial x + \partial (K_y (\partial C / \partial y)) / \partial y$$

and

$$(2.33) \quad \partial C / \partial t = + \partial (K_x (\partial C / \partial x)) / \partial x.$$

The equations (2.23), (2.32) and (2.33) can sometimes be simplified. If, for example, an assumption similar to (2.24) is made, by using some constants K_x and K_y , then (2.32) could be rewritten as

$$(2.34) \quad \partial C / \partial t = + K_x \partial^2 C / \partial x^2 + K_y \partial^2 C / \partial y^2.$$

Some more details about the diffusion equations will be given in the following chapters where it will be explained how these equations can be treated numerically.

There are still many open questions related to the mathematical description of the diffusion phenomena. The description given above is a rather crude approximation. Since **the numerical treatment** of large mathematical models is the main topic in this book, there is no need to discuss this topic in further details. It should be emphasized, however, that it will be possible to treat numerically many other descriptions of the diffusion process with the same numerical methods as those that will be described in the following chapter.

2.5. Modelling the deposition processes

Both dry and wet deposition processes take place during the long-range transport of air pollutants. The dry deposition takes place throughout the transport, while the wet deposition takes place only when some kind of precipitation occurs in the area under consideration.

The same approach as in the previous two sections can be used. Let us assume that deposition only takes place (i.e. there is no advection, no diffusion, no emission and no chemistry). Again, as in the previous two sections, this is an artificial situation, because the deposition phenomena should be considered together with the other four basic physical processes. However, the study of the deposition only has several advantages (as already mentioned in the previous two sections where advection only and diffusion only were studied):

(i) one can simplify the explanation of the main features of the deposition processes

and

(ii) one can facilitate the construction of the global mathematical model for studying long-range transport of air pollutants.

It is clear that the deposition part of a mathematical LRTAP model depends on the particular air pollutants that are transported (this was not the case for the first two physical processes; see the previous two sections). However, under certain assumptions the deposition of a given pollutant does not depend on the deposition of the other pollutants involved in the mathematical model. If such an assumption is made, then the deposition depends only on the appropriate concentration (not on its spatial derivatives). This implies that ordinary differential equations can be used in this situation (instead of partial differential equations). Moreover, it is sufficient to consider only one ordinary differential equation in order to explain the main mechanisms concerning the deposition processes (dry and wet deposition), but it should be reiterated that this is only true when some appropriate assumptions are imposed. Thus, although the global mathematical LRTAP model is described by a system of PDE's (the number of the equations in the system being equal to the number of air pollutants involved in the study),

in some cases it is quite sufficient to consider a single ordinary differential equation (an ODE) in order to explain the deposition processes (note that this is different from the previous two physical processes, where a partial differential equation, a PDE, was considered).

Let us denote, as in the previous section, the concentration of a given air pollutant at the point $(x,y,z) \in D \subset \mathbf{R}^3$ at time $t \in [0,T]$ by $c_s(x,y,z,t)$, $s=1,2,\dots,q$ (q being the number of pollutants studied by the model), where D is a space domain in the three-dimensional Euclidean space $\mathbf{R}^3$ and $[0,T]$ is a time-interval with some $T > 0$. Let us denote the dry deposition coefficient with $k_{1s}(x,y,t)$ and the wet deposition coefficient with $k_{2s}(x,y,z,t)$, $s=1,2,\dots,q$ (note that the dry deposition coefficient does not depend on z). Furthermore, let us also assume that both the dry deposition coefficient and the wet deposition coefficient are known functions defined in the space domain D and on the time-interval $[0,T]$.

If there is no danger for misunderstanding (see also the previous two sections), then the independent variables x , y , z and t will as a rule be omitted; i.e. the abbreviations c_s , k_{1s} and k_{2s} will be used instead of $c_s(x,y,z,t)$, $k_{1s}(x,y,t)$ and $k_{2s}(x,y,z,t)$, respectively.

When large LRTAP models are considered, the space domain is normally a parallelepiped in the three dimensional Euclidean space $\mathbf{R}^3$. If an assumption is made that the space domain is a parallelepiped, then the space domain D can be defined, as in the previous two sections, by (2.1).

Under the assumptions made above, the deposition process can be described by the following linear ordinary differential equation (it should be mentioned here that this equation will differ in the transition from one air pollutant to another, but when the air pollutant is fixed it depends on the fixed air pollutant only and not on the others):

$$(2.35) \quad dc_s/dt = - (k_{1s} + k_{2s}) c_s.$$

As mentioned above, both k_{1s} and k_{2s} will generally depend on both of the spatial coordinates and on the time variable. However, for some species these coefficients could be assumed to be constants. In both cases (when k_{1s} and k_{2s} depend on x , y , z and t and when k_{1s} and k_{2s} are constants), the equation (2.35) is linear and can be solved exactly.

In some cases an assumption that the deposition coefficients k_{1s} and k_{2s} depend on the concentration c_s (at least for some values of s ; $s=1,2,\dots,q$) might be made. If this is the case, then (2.35) becomes non-linear (at least for some values of s) and, thus, has to be treated numerically. As an illustration consider the wet deposition coefficient. One might assume that the rain drops cannot take more than a certain number of particles of a given pollutant. This means that the wet deposition coefficient k_{2s} will depend not only on x , y , z and t , but also on c_s . If this is the case, then some of the terms in the right-hand-side of (2.35) will no longer be linear in c_s . It should be noted here that there seems to be some evidence that such an assumption is not very unrealistic: it could be confirmed experimentally (for sulphur pollutants, at least) by comparing the concentrations of sulphate in precipitation with the concentrations of sulphate in air.

The most complicated case arises when one has to assume that the deposition of a given pollutant c_s ($s=1,2,\dots,q$, where q is the number of the pollutants studied) depends not only on the concentration of c_s but also on the concentrations of some other pollutants. In this situation the deposition process cannot be studied by considering a single equation of the type

(2.35). A system of ordinary differential equations is to be solved in this case. Such a system of ODE's can be written as

$$(2.36) \quad \frac{dc_s}{dt} = S_s(c_1, c_2, \dots, c_q), \quad s=1, 2, \dots, q.$$

This system is very similar to the system of ODE's that arises in the chemical part of the model (when the chemical reactions are described mathematically; see the next two sections).

More details about the mathematical description of the deposition process will be given in the following chapters.

2.6. Choice of a chemical scheme

One of the most important physical processes, which takes place under the transport of pollutants in the atmosphere and which should be treated very carefully, is the chemistry: chemical transformations take place during the long-range transport of air pollutants. It is highly desirable to determine an optimal set of chemical reactions and to implement it in the model (this issue has been discussed in Section 1.5). Unfortunately this problem (the determination of an optimal set of chemical reactions) has until now been solved only in some very special cases; as, for example, in the case where long-range transport phenomena concerning only sulphur pollutants were studied. In more general cases (where several different types of air pollutants are to be included in the model), it is rather difficult to solve the problem and, therefore, many more research efforts in this important direction are highly desirable.

Assume that most of the main air pollutants (sulphur pollutants, nitrogen pollutants, ozone and hydro-carbons) are to be studied. Then the so-called condensed CBM IV scheme¹, developed by Gery et al. [204], can (in some cases, at least) be used successfully. The chemical reactions that are involved in the condensed CBM IV scheme are listed in Table 2.3. Several chemical reactions involving ammonia-ammonium have been added to this scheme. The treatment of these reactions is based on the algorithm proposed in Iversen et al. [252-254]. The condensed CBM IV chemical scheme is still used in **the Danish Eulerian Model**. However, some work is carried now to incorporate several more advanced chemical schemes in this model. This is a common tendency: the scientists are trying to develop and incorporate in the large air pollution model more advanced (and, hopefully, more reliable) chemical schemes.

¹ **CBM**: Carbon Bond Mechanism.

No.	Chemical reaction	Coefficient	Exponent
1	$\text{NO}_2 + h\nu \Rightarrow \text{NO} + \text{O}$	$1.00 \cdot 10^{-2}$	0.39
2	$\text{O} \Rightarrow \text{O}_3$	$4.65 \cdot 10^{-6}$	- 690
3	$\text{O}_3 + \text{NO} \Rightarrow \text{NO}_2$	$2.66 \cdot 10^{-1}$	1490
4	$\text{O} + \text{NO}_2 \Rightarrow \text{NO}$	$1.38 \cdot 10^{-4}$	
5	$\text{O} + \text{NO}_2 \Rightarrow \text{NO}_3$	$2.32 \cdot 10^{-3}$	- 600
6	$\text{O} + \text{NO} \Rightarrow \text{NO}_2$	$3.21 \cdot 10^{-3}$	- 411
7	$\text{NO}_2 + \text{O}_3 \Rightarrow \text{NO}_3$	$4.74 \cdot 10^{-2}$	2450
8	$\text{O}_3 + h\nu \Rightarrow 0.205 \cdot \text{OH} + 0.897 \cdot \text{O}$	$1.90 \cdot 10^{-4}$	1.90
9	$\text{O}_3 \Rightarrow \text{O}$	$4.20 \cdot 10^{-2}$	
10	$\text{O}_3 + \text{OH} \Rightarrow \text{HO}_2$	$1.00 \cdot 10^{-2}$	940
11	$\text{O}_3 + \text{HO}_2 \Rightarrow \text{OH}$	3.00	580
12	$\text{NO}_3 + \text{NO} \Rightarrow 2 \cdot \text{NO}_2$	$2.81 \cdot 10^{-4}$	- 250
13	$\text{NO}_3 + \text{NO}_2 \Rightarrow \text{NO} + \text{NO}_2$	$5.90 \cdot 10^{-1}$	1230
14	$\text{NO}_3 + \text{NO}_2 \Rightarrow \text{N}_2\text{O}_5$	$1.78 \cdot 10^{-3}$	60
15	$\text{N}_2\text{O}_5 \Rightarrow 2 \cdot \text{NO}_3$	$2.85 \cdot 10^{-2}$	
16	$\text{N}_2\text{O}_5 \Rightarrow \text{NO}_3 + \text{NO}_2$	3.12	10840
17	$\text{NO} + \text{NO}_2 \Rightarrow 2 \cdot \text{HNO}_2$	$2.40 \cdot 10^{-7}$	
18	$\text{HNO}_2 + \text{HNO}_2 \Rightarrow \text{NO} + \text{NO}_2$	$1.50 \cdot 10^{-5}$	
19	$\text{HNO}_2 + h\nu \Rightarrow \text{NO} + \text{OH}$	$1.80 \cdot 10^{-3}$	0.39
20	$\text{NO}_2 + \text{OH} \Rightarrow \text{HNO}_3$	$1.63 \cdot 10^{-4}$	- 560
21	$\text{NO} + \text{OH} \Rightarrow \text{HNO}_2$	$9.77 \cdot 10^{-3}$	- 477
22	$\text{HO}_2 + \text{NO} \Rightarrow \text{OH} + \text{NO}_2$	$1.23 \cdot 10^{-4}$	- 240
23	$\text{NO} + \text{NO} \Rightarrow 2 \cdot \text{NO}_2$	$1.52 \cdot 10^{-4}$	- 530
24	$\text{OH} + \text{HNO}_2 \Rightarrow \text{NO}_2$	$9.77 \cdot 10^{-3}$	
25	$\text{NO}_3 + h\nu \Rightarrow 0.85 \cdot \text{NO}_2 + 0.85 \cdot \text{O} + 0.15 \cdot \text{NO}$	$3.06 \cdot 10^{-1}$	0.39
26	$\text{HO}_2 + \text{HO}_2 \Rightarrow \text{H}_2\text{O}_2$	$4.144 \cdot 10^{-3}$	-1150
27	$\text{HO}_2 + \text{HO}_2 \Rightarrow \text{H}_2\text{O}_2$	$3.27 \cdot 10^{-3}$	-5800
28	$\text{OH} + \text{CO} \Rightarrow \text{HO}_2$	$4.00 \cdot 10^{-2}$	
29	$\text{FORM} + \text{OH} \Rightarrow \text{HO}_2 + \text{CO}$	$1.50 \cdot 10^{-4}$	
30	$\text{FORM} + h\nu \Rightarrow 2 \cdot \text{HO}_2 + \text{CO}$	$7.80 \cdot 10^{-5}$	0.87
31	$\text{FORM} + h\nu \Rightarrow \text{CO}$	$1.10 \cdot 10^{-4}$	0.59
32	$\text{FORM} + \text{O} \Rightarrow \text{OH} + \text{HO}_2 + \text{CO}$	$2.37 \cdot 10^{-2}$	1550
33	$\text{FORM} + \text{NO}_3 \Rightarrow \text{HO}_2 + \text{CO}$	$9.30 \cdot 10^{-1}$	
34	$\text{ALD}_2 + \text{O} \Rightarrow \text{C}_2\text{O}_3 + \text{OH}$	$6.36 \cdot 10^{-2}$	986

No.	Chemical reaction (continuation)	Coefficient	Exponent
35	$\text{ALD}_2 + \text{O} \Rightarrow \text{C}_2\text{O}_3$	$2.40 \cdot 10^{-4}$	- 250
36	$\text{ALD}_2 + \text{NO}_3 \Rightarrow \text{C}_2\text{O}_3$	3.70	
37	$\text{ALD}_2 + h\nu \Rightarrow \text{XO}_2 + 2 \cdot \text{HO}_2 + \text{CO} + \text{FORM}$	$1.50 \cdot 10^{-5}$	1490
38	$\text{ALD}_2 + \text{HO}_2 \Rightarrow \text{XO}_2 + \text{HO}_2 + 2 \cdot \text{FORM}$	5.00	
39	$\text{C}_2\text{O}_3 + \text{NO} \Rightarrow \text{NO}_2 + \text{XO}_2 + \text{FORM} + \text{HO}_2$	$1.65 \cdot 10^{-4}$	- 250
40	$\text{C}_2\text{O}_3 + \text{NO}_2 \Rightarrow \text{PAN}$	$9.00 \cdot 10^{-3}$	- 250
41	$\text{PAN} \Rightarrow \text{C}_2\text{O}_3 + \text{NO}_2$	$2.22 \cdot 10^{-2}$	14400
42	$\text{C}_2\text{O}_2 + \text{C}_2\text{O}_3 \Rightarrow 2 \cdot (\text{XO}_2 + \text{FORM} + \text{HO}_2)$	$3.70 \cdot 10^{-3}$	
43	$\text{C}_2\text{O}_3 + \text{HO}_2 \Rightarrow \text{the result is not used}$	$9.63 \cdot 10^{-3}$	
44	$\text{MGLY} \Rightarrow \text{C}_2\text{O}_3 + \text{HO}_2 + \text{CO}$	$2.00 \cdot 10^{-4}$	
45	$\text{MGLY} + \text{OH} \Rightarrow \text{XO}_2 + \text{C}_2\text{O}_3$	$2.64 \cdot 10^{-4}$	
46	$\text{OH} \Rightarrow \text{XO}_2 + \text{FORM} + \text{HO}_2$	$2.10 \cdot 10^{-1}$	
47	$\text{PAR} \Rightarrow 1.49 \cdot \text{XO}_2 + 0.067 \cdot \text{XO}_2\text{N} + 0.93 \cdot \text{HO}_2 + 0.45 \cdot \text{ALD}_2 - 0.75 \cdot \text{PAR}$	$1.15 \cdot 10^{-3}$	
48	$\text{O} + \text{OLE} \Rightarrow 0.95 \cdot \text{ALD}_2 + 0.35 \cdot \text{HO}_2 + 0.2 \cdot \text{XO}_2 + 0.15 \cdot \text{CO} + 0.05 \cdot \text{FORM} + 0.05 \cdot \text{C}_2\text{O}_3 - 0.35 \cdot \text{PAR}$	$5.92 \cdot 10^{-3}$	324
49	$\text{OH} + \text{OLE} \Rightarrow \text{FORM} + \text{ALD}_2 + \text{XO}_2 + \text{HO}_2 - \text{PAR}$	$4.20 \cdot 10^{-4}$	- 537
50	$\text{O}_3 + \text{OLE} \Rightarrow 0.5 \cdot \text{ALD}_2 + 0.66 \cdot \text{FORM} + 0.212 \cdot \text{CO} + 0.28 \cdot \text{HO}_2 + 0.144 \cdot \text{XO}_2 + 0.08 \cdot \text{OH} - \text{PAR}$	$1.80 \cdot 10^{-2}$	1897
51	$\text{NO}_3 + \text{OLE} \Rightarrow 0.91 \cdot (\text{XO}_2 + \text{HO}_2) + 0.09 \cdot \text{XO}_2\text{N}$	$1.14 \cdot 10^{-1}$	
52	$\text{O} + \text{ETH} \Rightarrow \text{FORM} + \text{XO}_2 + \text{CO} + 2 \cdot \text{HO}_2$	$1.08 \cdot 10^{-3}$	800
53	$\text{OH} + \text{ETH} \Rightarrow \text{XO}_2 + 2 \cdot \text{FORM} + \text{HO}_2$	$1.20 \cdot 10^{-4}$	- 320
54	$\text{O}_3 + \text{ETH} \Rightarrow \text{FORM} + 0.37 \cdot \text{CO} + 0.13 \cdot \text{HO}_2$	$2.70 \cdot 10^{-3}$	2840
55	$\text{TOL} + \text{OH} \Rightarrow \text{HO}_2 + 0.64 \cdot \text{XO}_2 + 1.13 \cdot \text{FORM} + 0.56 \cdot \text{MGLY} + 0.36 \cdot \text{PHEN} + 0.36 \cdot \text{PAR} + 1.13 \cdot \text{CO}$	$9.75 \cdot 10^{-3}$	
56	$\text{PHEN} + \text{NO}_3 \Rightarrow \text{PHO}$	$1.40 \cdot 10^{-4}$	
57	$\text{PHO} + \text{HO}_2 \Rightarrow \text{the result is not used}$	$2.00 \cdot 10^{-4}$	
58	$\text{XYL} + \text{OH} \Rightarrow \text{HO}_2 + 0.72 \cdot \text{O}_2 + 0.67 \cdot \text{CO} + 1.33 \cdot \text{MGLY} + 0.28 \cdot \text{PHEN} + 0.67 \cdot \text{FORM} + 0.56 \cdot \text{PAR}$	$3.60 \cdot 10^{-4}$	
59	$\text{XO}_2\text{N} + \text{NO} \Rightarrow \text{the result is not used}$	$1.00 \cdot 10^{-3}$	
60	$\text{XO}_2 + \text{NO} \Rightarrow \text{NO}_2$	$1.20 \cdot 10^{-4}$	
61	$\text{XO}_2 + \text{HO}_2 \Rightarrow \text{the result is not used}$	$5.00 \cdot 10^{-3}$	-1300
62	$\text{XO}_2 + \text{C}_2\text{O}_3 \Rightarrow \text{XO}_2 + \text{HO}_2 + \text{FORM}$	$2.40 \cdot 10^{-3}$	

No.	Chemical reaction (continuation)	Coefficient	Exponent
63	$\text{SO}_2 + \text{OH} \Rightarrow \text{HO}_2 + \text{H}_2\text{SO}_4$	$1.50 \cdot 10^{-3}$	
64	$\text{SO}_2 \Rightarrow \text{H}_2\text{SO}_4$	$8.30 \cdot 10^{-5}$	
65	$\text{HO}_2 + \text{HO}_2 \Rightarrow \text{H}_2\text{O}_2$	$2.20 \cdot 10^{-13}$	
66	$\text{H}_2\text{O}_2 + h\nu \Rightarrow 2 \cdot \text{OH}$	$1.73 \cdot 10^{-5}$	0.767
67	$\text{H}_2\text{O}_2 + \text{OH} \Rightarrow \text{HO}_2 + \text{H}_2\text{O}$	$2.90 \cdot 10^{-2}$	-160
68	$\text{HNO}_3 \Rightarrow \text{NO}_3^-$	$1.00 \cdot 10^{-5}$	
69	$\text{NO}_3^- \Rightarrow \text{HNO}_3$	$5.00 \cdot 10^{-6}$	14400
70	$\text{H}_2\text{SO}_4 + \text{NH}_3 \Rightarrow (\text{NH}_4)_{1.5}\text{SO}_4$	∞	
71	$\text{HNO}_3 + \text{NH}_3 \rightleftharpoons \text{NH}_4\text{NO}_3$	k_{70}	

Table 2.3
Chemical reactions from the CBM IV scheme
together with some ammonium-ammonia reactions.

The condensed CBM IV scheme is rather complicated and, therefore, it is useful to have a simpler scheme when the most important features of the numerical treatment of the chemical scheme are to be studied. Two main requirements must be satisfied by such a simple scheme: (i) it should allow us to see readily the main trends in the variations of the concentrations in the most interesting cases and (ii) it should still contain all important types of chemical reactions. Such a simple chemical scheme, a simplified chemical scheme, is presented in Table 2.4.

The simplified chemical scheme will be used in most of the numerical experiments. It is rather simple (only 10 species are involved in this scheme), but all types of chemical reactions that are difficult for the numerical methods (non-linear chemical reactions² and photochemical reactions³) are represented in it. Therefore numerical methods that perform well in the treatment of this scheme should be expected to perform also well in the treatment of more complicated chemical schemes (as the scheme given in Table 2.3).

² The non-linear chemical reactions are described mathematically by terms that depend non-linearly on the involved concentrations. In the cases considered here, the terms by which a non-linear reaction is described will contain products of two concentrations; see the mathematical description of the simplified chemical scheme given by (2.51) - (2.60). If a chemical reaction can be described mathematically by terms that depend only linearly on the involved concentrations, then it will be called linear.

³ In photochemical reactions the initiating step is absorption of a photon by an atom, molecule, free radical or ion. To be initiated photochemical reactions require light in the visible and ultraviolet portion of the spectra. The absorption is often followed by dissociation. The initial effect of photon absorption and dissociation is called a **primary photochemical reaction**. Subsequent reactions caused by products from the primary photochemical reactions are called **secondary photochemical reactions**. Both the primary and the secondary photochemical reactions are important. More details can be found in Calvert and Pitts [58] (see also [100] and [103]).

No.	Chemical reaction	Coefficient	Exponent
1	$\text{HC} + \text{OH} \Rightarrow 4\text{RO}_2 + 2\text{ALD}$	$6.00 \cdot 10^{-12}$	
2	$\text{ALD} + h\nu \Rightarrow 2\text{HO}_2 + \text{CO}$	$7.80 \cdot 10^{-5}$	0.87
3	$\text{RO}_2 + \text{NO} \Rightarrow \text{NO}_2 + \text{ALD} + \text{HO}_2$	$8.00 \cdot 10^{-12}$	
4	$\text{HO}_2 + \text{NO} \Rightarrow \text{NO}_2 + \text{OH}$	$8.00 \cdot 10^{-12}$	
5	$\text{NO}_2 + h\nu \Rightarrow \text{NO} + \text{O}_3$	$1.00 \cdot 10^{-2}$	0.39
6	$\text{NO} + \text{O}_3 \Rightarrow \text{NO}_2 + \text{O}_2$	$1.60 \cdot 10^{-14}$	
7	$\text{O}_3 + h\nu \Rightarrow \text{O}_2 + \text{O}(^1\text{D})$	$1.90 \cdot 10^{-4}$	1.90
8	$\text{O}(^1\text{D}) + \text{H}_2\text{O} \Rightarrow 2\text{OH}$	$2.30 \cdot 10^{-10}$	
9	$\text{NO}_2 + \text{OH} \Rightarrow \text{HNO}_3$	$1.00 \cdot 10^{-11}$	
10	$\text{CO} + \text{OH} \Rightarrow \text{CO}_2 + \text{HO}_2$	$2.90 \cdot 10^{-13}$	

Table 2.4
Chemical reactions from the simplified chemical scheme
(HC is a lumped hydro-carbon molecule, ALD is formaldehyde).

A short description of the notation used in Table 2.3 and Table 2.4 is needed. Such a description is given below.

2.6.1. Chemical reactions that are not photochemical

If the chemical reaction under consideration is not photochemical, i.e. there is no term $h\nu$ in the expression for the reaction given in Table 2.3 or in Table 2.4, then the rate coefficients (as the name indicates, the speed of the chemical reactions is measured in some sense by their rate coefficients) are of the form:

$$(2.37) \quad k = A \exp(-B/T),$$

where

(i) the parameter T is the temperature (which is measured in degrees Kelvin),

(ii) the parameter B is the Arrhenius temperature coefficient for the reaction under consideration (which is a constant measured in degrees Kelvin)

and

(iii) the parameter A depends on the particular chemical reaction under consideration (the units used are the same as those used in connection with k ; see Sub-section 2.6.3).

The values of all rate coefficients k at 298K are listed in the third column of the table. The parameters B (when these are non-zeros) are listed in the fourth column of the table. Thus, the parameters A can be determined by using (2.37) and the figures listed in the table (for the non-photochemical reactions).

It should be mentioned that the description given here is in fact simplified; the rate coefficients of the chemical reactions may also depend on some other parameters (as, for example, the pressure). It is believed that the description given above works rather satisfactorily in many practical situations. Therefore it is often used in large air pollution models.

2.6.2. Photochemical reactions

For the photochemical reactions (i.e. for all chemical reactions in Table 2.3 that contain the term $h\nu$; these reactions depend on the light and, thus, on the position of the Sun relative to the horizon) the rate coefficients are given by a formula of the following type:

$$(2.38) \quad k = A \exp(-B/\cos\theta),$$

where θ is the solar elevation. It should be noted that the rate coefficients for the photochemical reactions are not given in the table: the figures in the table are the values of A and B from (2.38) when the reaction is photochemical.

2.6.3. The units used

The rate coefficients k are measured in $1/\text{min}$ for unimolecular reactions as well as for the photochemical reactions (the reactions containing the term $h\nu$) and in $1/(\text{ppm min})$ for all bimolecular reactions. It is clear that the parameters A are measured with the same units as the rate coefficients k . These units are often transformed in the models; in order to achieve compatibility with the other units that are used in the models.

For some reactions the terms in the right-hand side are not given. This means that the final result of such a reaction consists of harmless final products that are not needed further on in the model.

2.6.4. The ammonia-ammonium chemistry

The ammonia-ammonium chemistry (that is used in connection with the condensed CBM IV chemical scheme) is based on simple equilibrium relationships.

It is assumed that the ammonia, NH_3 , reacts first with sulphate. It is furthermore assumed that the reaction takes place instantly and the compound (either NH_3 or sulphate) whose concentration is smaller is totally absorbed in the $(\text{NH}_4)_{1.5}\text{SO}_4$.

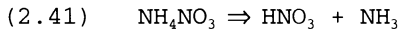
The description of the last reaction in Table 2.3 table is much more complicated. In the description given below the main ideas described in Iversen et al. [252-254] have been followed. Let $[z]$ denote the concentration of z . A quantity x is determined by the use of the equilibrium relationship

$$(2.39) \quad ([\text{NH}_3] + x)([\text{HNO}_3] + x) = k_{70}$$

and by taking the root of the quadratic equation (2.39) that has physical meaning:

$$(2.40) \quad x = 0.5 \{ -([\text{NH}_3] + [\text{HNO}_3]) + ([\text{NH}_3] - [\text{HNO}_3])^2 + 4k_{70} \}^{0.5}.$$

If x is positive and $[\text{NH}_4\text{NO}_3] - x$ is non-negative, then the chemical reaction



takes place and the new concentrations are

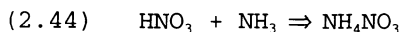
$$(2.42) \quad [\text{NH}_4\text{NO}_3] - x, \quad [\text{NH}_3] + x \quad \text{and} \quad [\text{HNO}_3] + x.$$

If x is positive and $[\text{NH}_4\text{NO}_3] - x$ is negative, then again the reaction (2.41) takes place, but the new concentrations are

$$(2.43) \quad [\text{NH}_4\text{NO}_3] - y, \quad [\text{NH}_3] + y \quad \text{and} \quad [\text{HNO}_3] + y,$$

where $y = [\text{NH}_4\text{NO}_3]$. It should be noted that the equilibrium is not achieved in this situation.

If x is negative, then it can be proved that both $[\text{NH}_3] + x$ and $[\text{HNO}_3] + x$ are non-negative and the reaction



takes place. The new concentrations are again given by (2.40).

It is clear that if $x = 0$, then nothing happens (i.e. the concentrations involved, $[\text{NH}_4\text{NO}_3] - x$, $[\text{NH}_3] + x$ and $[\text{HNO}_3] + x$, remain unchanged).

To complete the description of the mechanism, it is necessary to define the coefficient k_{70} which is used in (2.40). Denote by HUM the relative humidity and let c_d be defined by

$$(2.45) \quad \ln(c_d) = 856.23/T + 1.2306.$$

For $\text{HUM} < c_d$, the coefficient k_{70} from (2.40), can be calculated by using the following relationship:

$$(2.46) \quad \ln(k_{70}) = 70.78 - 24.220/T - 6.1 \ln(T/298),$$

Consider now the case $\text{HUM} \geq c_d$. Let k^* be determined by

$$(2.48) \quad \ln(k^*) = 70.78 - 24.200/T - 6.1 \ln(T/298) .$$

Then the coefficient k_{70} from (2.40) can be calculated from the following equality:

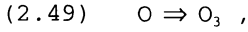
$$(2.47) \quad \ln(k_{70}) = \ln(k^*) - (20.75 - \ln(k^*)) (c - c_d) / (101 - c_d)$$

when either $\text{HUM} > c_d$ or $\text{HUM} = c_d$.

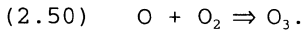
Thus, the chemical mechanism for the reactions involving NH_4SO_4 and NH_4NO_3 is based on an assumption of equilibrium relationships. This mechanism is fully described in Iversen et al. [252]-[254]. An attempt to improve this mechanism is discussed and tested (comparing the model results with measurement from the North Sea taken from a ship) in Harrison et al. [228].

2.6.5. Simplifying the notation

The following assumption is made to simplify the notation used in Table 2.3: all compounds that are not important for the model are not included in the reactions. Thus, the second reaction in Table 2.3 is written as



while, strictly speaking, it is more correct to apply the somewhat longer formula



This means that the oxygen molecule O_2 is omitted in Table 2.3 (because the oxygen is not an air pollutant and the above reaction does not lead to any practical change of the oxygen in the atmosphere). In this way the notation in Table 2.3 is simplified; the hope is that the contents of the table are also quite clear when such a simplification is performed.

2.7. Mathematical description of the chemical reactions

The chemical reactions can be described mathematically by a system of ordinary differential equations (ODE's). For the small chemical scheme with ten species, see Table 2.4, the system of ODE's can be described as follows.

Let us assume that the mathematical notation that is shown in Table 2.5 is used in connection with the ten chemical species from the simplified chemical scheme.

By applying this notation, the simplified chemical scheme can easily be represented by the following system of ODE's (one equation per every chemical species).

$$(2.51) \quad d\eta_1/dt = k_5\eta_2 - (k_3\eta_8 + k_4\eta_7 + k_6\eta_5) \eta_1$$

$$(2.52) \quad d\eta_2/dt = (k_3\eta_8 + k_4\eta_7 + k_6\eta_5) \eta_1 - (k_5 + k_9\eta_9) \eta_2$$

$$(2.53) \quad d\eta_3/dt = -k_1\eta_9\eta_3$$

$$(2.54) \quad d\eta_4/dt = 2k_1\eta_9\eta_3 + k_3\eta_8\eta_1 - k_2\eta_4$$

$$(2.55) \quad d\eta_5/dt = k_5\eta_2 - k_6\eta_5\eta_1 - k_8h\eta_8$$

$$(2.56) \quad d\eta_6/dt = k_9\eta_9\eta_2$$

$$(2.57) \quad d\eta_7/dt = 2k_2\eta_4 + k_3\eta_8\eta_1 + c\eta_9 - k_4\eta_7\eta_1$$

$$(2.58) \quad d\eta_8/dt = 4k_1\eta_9 - k_3\eta_1\eta_8$$

$$(2.59) \quad d\eta_9/dt = k_4\eta_1\eta_7 + 2k_8h\eta_7 - (k_6\eta_3 + k_9\eta_2 - c)\eta_9$$

$$(2.60) \quad d\eta_{10}/dt = k_7\eta_5 - (k_8h + f)\eta_2$$

where c , h and f are appropriately defined constants. It is not necessary to explain in detail the meaning of these constants; the main purpose is to show that the systems of chemical reactions in the simplified chemical scheme can be represented mathematically by a system of ordinary differential equations.

Chemical notation	Mathematical notation
NO	η_1
NO ₂	η_2
HC	η_3
ALD	η_4
O ₃	η_5
HNO ₃	η_6
HO ₂	η_7
RO ₂	η_8
OH	η_9
O(¹ D)	η_{10}

Table 2.5
The mathematical notation that is used
in connection with the simplified
chemical scheme (10 species).

It is clear that the ODE system (2.51)-(2.60) can be rewritten in vector form as

$$(2.61) \quad d\eta/dt = Q(\eta),$$

where the vector η is given by $\eta = \{\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6, \eta_7, \eta_8, \eta_9, \eta_{10}\}^T$, while the components of vector Q are the right-hand side parts of the equalities (2.51)-(2.60).

The chemists and the physicists prefer to divide the right hand-sides of the equalities (2.51)-(2.60) into two parts:

$$(2.62) \quad d\eta_i/dt = P_i(\eta) + L_i(\eta) \eta_i, \quad i=1,2,\dots,10.$$

This division has some physical meaning: P_i is called the production term for the i 'th pollutant involved in the chemical scheme, while L_i is the loss term. However, it is important to emphasize here that both terms are generally non-linear.

The units normally used by chemists for the concentrations of the various species are molecules per cubic centimetre. Therefore the system of ODE's, (2.62), is very badly scaled. The situation becomes worse when the photochemical reactions are activated or deactivated: this leads to very quick changes of the concentrations in short time-periods. Both the bad scaling of the problem and the quick variations of the concentrations over short periods of time cause great difficulties when the chemical part of the model is to be treated numerically (this will be discussed in detail in the following chapters of this book).

The initial conditions together with the magnitudes of the minimal and maximal concentrations found during a short run (42 hours) of the simplified chemical scheme defined by (2.51)-(2.60) are shown in Table 2.6.

The system of ODE's corresponding to the condensed CBM IV chemical scheme, the scheme with 35 species, can be derived in a quite similar way.

The chemical schemes, and especially their numerical treatment will also be discussed in the following chapters. It should also be stressed here, however, that many more efforts are needed in order to find numerical procedures by which the chemical part of the LRTAP models can be treated efficiently. As mentioned several times in this section, and in the previous sections, it is crucial to treat the chemical part of the LRTAP models efficiently by fast, accurate and robust numerical methods (if this is not done, then very often the models cannot be treated numerically even if modern high-speed computers are available). It seems to be necessary to develop quite new integration methods for the treatment of the stiff and badly scaled systems of ODE's that arise from the chemical part of the air pollution models.

Pollutant	Initial condition	Minimal value	Maximal value
NO	$1.0 \cdot 10^{11}$	$7.1 \cdot 10^{-24}$	$1.0 \cdot 10^{11}$
NO ₂	$1.0 \cdot 10^{11}$	$7.6 \cdot 10^7$	$1.0 \cdot 10^{11}$
HC	$1.0 \cdot 10^{11}$	$8.4 \cdot 10^8$	$1.0 \cdot 10^{11}$
ALD	$5.0 \cdot 10^{10}$	$5.0 \cdot 10^{10}$	$5.1 \cdot 10^{11}$
O ₃	$5.0 \cdot 10^{11}$	$5.0 \cdot 10^{11}$	$1.6 \cdot 10^{12}$
HNO ₃	$1.0 \cdot 10^{10}$	$1.0 \cdot 10^{10}$	$2.1 \cdot 10^{11}$
HO ₂	$1.0 \cdot 10^6$	$1.0 \cdot 10^6$	$1.3 \cdot 10^{12}$
RO ₂	$1.0 \cdot 10^6$	$1.0 \cdot 10^6$	$4.3 \cdot 10^{10}$
OH	$1.0 \cdot 10^5$	$1.9 \cdot 10^{-19}$	$7.3 \cdot 10^7$
O(¹ D)	$1.0 \cdot 10^{-3}$	$3.4 \cdot 10^{-29}$	$2.6 \cdot 10^{-2}$

Table 2.6
The initial values and the order of magnitude
of the minimal and maximal concentrations
obtained in a short run
(starting at 6 o'clock in the morning
and finishing at 24 o'clock in the next day).

2.8. Introduction of the emissions in the models

The introduction of the emission sources from the space domain under consideration in the long-range transport models should be, theoretically at least, a straight-forward process. In practice, however, it is rather difficult to identify the sources and, what is sometimes even more important, to determine quantitatively the amount of pollution emitted from a given source. The practical problems connected with the introduction of the emission sources in large-scale air pollution models become even greater when the sources are time-dependent (which is often the case).

Assume again, for the sake of simplicity only, that a two-dimensional model is to be treated. Consider some primary pollutant whose concentration is $c_i(x,y,t)$. The positions of the sources of this primary pollutant (see Section 1.2) are to be located and the emission at a point (x,y) and at time t is to be described by a function $E_i(x,y,t)$. Function $E_i(x,y,t)$ takes non-negative values. In general, $E_i(x,y,t) > 0$ if the i 'th pollutant is a primary pollutant and if there is a source at point (x,y) . However, $E_i(x,y,t)$ may sometimes be zero even if there is a source at the point under consideration; for example, some sources may stop the emission during the night. If there is no source at point (x,y) or if the i 'th pollutant is a secondary pollutant, then $E_i(x,y,t)$ is equal to zero for all values of the time-variable t .

It is clear that the same considerations are valid for three dimensional models. It is important to emphasize here that three-dimensional models can successively be used to separate the contributions due to surface sources from the contributions due to elevated sources (as, for example, power plants with high chimneys) in some receptor areas within the space domain of the model.

The emission is normally treated together with the chemical reactions (the emission terms are simply added to the production terms in the chemical reactions; see Section 2.6). This means that, when the emissions in the space domain of the model under consideration are described mathematically, there are no more problems with the emissions (these are just treated in the chemical part of the model). The difficulty is, as mentioned above, to describe mathematically the emissions.

2.9. General mathematical description of an air pollution model

In the previous sections of this chapter it has been shown how the five major physical processes (advection, diffusion, deposition, chemical reactions and emission) can be described mathematically. A general mathematical model can easily be obtained now by combining the mathematical expressions that have been derived for the five physical processes:

$$\begin{aligned}
 (2.63) \quad \partial c_s / \partial t = & - \partial (u c_s) / \partial x - \partial (v c_s) / \partial y - \partial (w c_s) / \partial z \\
 & + \partial (K_x \partial c_s / \partial x) / \partial x + \partial (K_y \partial c_s / \partial y) / \partial y + \partial (K_z \partial c_s / \partial z) / \partial z \\
 & - (k_{1s} + k_{2s}) c_s(x, y, z, t) \\
 & + E_s(x, y, z, t) \\
 & + Q_s(c_1, c_2, \dots, c_q) , \\
 & s = 1, 2, \dots, q .
 \end{aligned}$$

The first three terms on the right-hand side of the above system of PDE's describe the advection process (see Section 2.3). The next three terms describe the diffusion (see Section 2.4). The two terms in the third line of (2.63) describe the deposition in the system (see Section 2.5). The first of these two terms, the term containing k_{1s} , is the contribution of the dry deposition, while the second term, the one containing k_{2s} , is the contribution of the wet deposition. The emission sources in the space domain, which were discussed in Section 2.8, are described in (2.63) by the term $E_s(x, y, z, t)$. Finally, the chemistry, the chemical reactions were studied in Section 2.6 and in Section 2.7, is represented in the model by the terms $Q_s(c_1, c_2, \dots, c_q)$: the terms $Q_s(c_1, c_2, \dots, c_q)$ are in fact the right-hand sides of the ordinary differential equations by which the chemical reactions are described mathematically (see Section 2.7).

It should be mentioned here that equations in system (2.63) are coupled only through the chemical terms. If the chemical terms $Q_s(c_1, c_2, \dots, c_q)$ were not present, then the system would split into independent equations. The chemical terms not only are coupling the equations in the system, but they also introduce non-linearity in the model.

The system (2.63) is to be considered together with appropriate initial and boundary conditions. The initial and boundary conditions will be discussed together with the numerical treatment of the model. However, it must be mentioned here that the dry deposition can be included in the boundary conditions (and this is often done).

Most of the coefficients in the system of PDE's are functions of both the space variables x, y, z and the time-variable t . Some of these functions depend on other parameters (as, for example, many of the rate coefficients that are used in the chemical schemes; these depend on the temperature, the humidity and other quantities). It is assumed that all these functions are known functions. It was shown in the beginning of this chapter that this assumption very often causes great practical difficulties. It was also discussed there how the difficulties can be alleviated.

2.10. Lagrangian models versus Eulerian models

The system of PDE's by which long-range transport of air pollution phenomena are described mathematically, see (2.63), cannot be solved analytically (unless some very unrealistic assumptions about the known functions involved in this system are made). Therefore (2.63) must be treated numerically. There are two main approaches that could be applied during the numerical treatment of (2.63): the Lagrangian approach and the Eulerian approach. Both approaches have successfully been used to handle many air pollution models numerically. The main properties of both the Lagrangian approach and the Eulerian approach will briefly be discussed in this section together with their advantages and disadvantages.

2.10.1. The Eulerian approach

Let us assume again, for the sake simplicity only, that a two-dimension model is to be handled. Assume also that the Eulerian approach is to be applied in the numerical treatment of the model. In such a case the space domain, on which the system of PDE's (2.63) is defined, is divided into "small" squares⁴ by applying some grid (equidistant grids are commonly used in air pollution modelling). Then the spatial derivatives involved in the

⁴ The grid-squares are small in the sense that the area of each square is a small part of the area of the whole space domain. In the old version of the Danish Eulerian Model the area of each grid square is $150 \text{ km} \times 150 \text{ km} = 22500 \text{ km}^2$, while the area of the space domain is $4800 \text{ km} \times 4800 \text{ km} = 23040000 \text{ km}^2$. This means that the area of one grid-square is approximately 0.0977% of the area of the space domain. Thus, although a $150 \text{ km} \times 150 \text{ km}$ square seems to be rather large (when it is considered as an independent object), it is small in connection with the whole space domain.

system of PDE's are discretized on the grid chosen. This is why the Eulerian models are sometimes also called grid models. After the discretization, the system of PDE's is normally reduced to several large systems of ODE's that are to be solved successively at every time-step. Different numerical algorithms can be applied in the computer treatment of these systems of ODE's.

In principle at least, it is also possible to discretize both the spatial derivatives and the time derivative. The fully discretized system, which is obtained from (2.63) when such a procedure is used, is a very large system of algebraic equations (the system of algebraic equations will be non-linear when non-linear chemical reactions are involved in the model). While the principle (on which the full discretization is based) is rather simple, it is very difficult to apply it to large air pollution models. Therefore the full discretization of the system of PDE's (2.63) is practically never used when large Eulerian air pollution models are treated numerically.

The numerical treatment of Eulerian models that are described mathematically by (2.63) will be discussed in Chapter 3.

2.10.2. The Lagrangian approach

The Lagrangian approach is based on the calculation of wind trajectories and on the transportation of air parcels along these trajectories. The models that are based on this approach are called either Lagrangian models or trajectory models. The main ideas used when a Lagrangian model is developed can be summarized as follows.

Consider a fixed point B in the space domain. A trajectory τ which arrives at B is calculated, and this trajectory can be followed backward for a certain time-period to reach some starting point A . A period of 96 hours is commonly used in the air pollution models; see, for example, Eliassen and Saltbones [154], Iversen et al. [252-254] or Simpson [413-414]. It is also possible to reverse the direction in which the trajectory is followed or, more precisely, to consider the situation where the starting point A is fixed and one follows forward the trajectory for a certain period (again periods of 96 hours are often used) until a final point B is reached. Thus, one can consider both backward and forward trajectories.

The numerical treatment of both the backward trajectories and the forward trajectories is the same, but the interpretation of the calculated results, when these two manners of modelling the long-range transport of air pollutants are applied, is as a rule different. In some particular situations it will be more convenient to use forward trajectories, while the application of backward trajectories will be the best solution in many other cases.

Let us assume (some of the assumptions listed below can be avoided if special devices are attached to the Lagrangian model under consideration) that

- (i) the concentrations of all air pollutants that are involved in the air pollution model under consideration are known at the starting point A (or,

at least, it is possible to obtain sufficiently accurate approximations to all concentrations of the air pollutants at the starting point A),

(ii) there is no diffusion in the air pollution process that is modelled

and

(iii) a two-dimensional model is to be developed (which implies that the trajectory τ is a two-dimensional curve).

Then the concentrations of the pollutants involved in the model at the final point B can be calculated by following the transport of the air parcel along the trajectory τ from point A to point B and by taking into account

(a) all contributions from the emission sources along the trajectory to the appropriate concentrations,

(b) all changes of the concentrations under the transport that are due to both the dry deposition process and the wet deposition process

and

(c) all transformations of the concentrations under the transport that are caused by the chemical reactions involved in the model.

The calculations (along the trajectory τ) can be carried out by using the following system of equations:

$$\begin{aligned}
 (2.64) \quad \frac{dc_s}{dt} = & - (k_{1s} + k_{2s}) c_s(x, y, t) \\
 & + E_s(x, y, t) + Q_s(c_1, c_2, \dots, c_q) , \\
 & s = 1, 2, \dots, q .
 \end{aligned}$$

It is immediately seen that a system of ODE's has to be treated when a Lagrangian model is used, while a system of PDE's, the system given by (2.63), has to be handled when an Eulerian model is applied. Thus, the system (2.63) is directly treated (by various numerical methods) when an Eulerian model is chosen. The system (2.63) is first simplified to the system (2.64) when a Lagrangian model is selected. It is important to emphasize here that this simplification is not obtained for nothing. The three assumptions (i) - (iii) are very often introduced if a Lagrangian model is to be used. Then the numerical treatment of system of ODE's (2.64), by which a Lagrangian model is described mathematically, is much easier than the numerical treatment of the corresponding system of PDE's (2.63), by which an Eulerian model is described mathematically. On the other hand, the assumptions (i) - (iii),

which are used to derive system (2.64) from system (2.63), may be rather restrictive in some situations.

2.10.3. Two ways of using the Lagrangian models

(A) Simple use of the Lagrangian models. The Lagrangian models are best suited for calculations in some simple (but very often important) situations as, for example, to answer the following question: What will happen if air parcels from the continent arrive at the Western coast of England passing over London? This situation has been studied by Derwent [101]. Note that forward trajectories can be used in this situation. This simplifies the problem: it is easier to calculate initial concentrations when forward trajectories are used, because the starting point of the trajectories is fixed and the concentrations at this point can, for example, be measured.

(B) Advanced use of the Lagrangian models. The truth is, however, that Lagrangian models are often used to produce the same output data files as the output data files that are calculated by the Eulerian models. This means that Lagrangian models are used to obtain values of the concentrations on the same types of grids as those used in the numerical treatment of Eulerian models. It is easy (in principle, at least) to solve the task of using a Lagrangian model in order to obtain the same output as that produced by an Eulerian model.

Consider any grid that has been chosen for an Eulerian model and let some initial value t_0 of the time variable t be fixed. Assume also that the number of grid-points in the space domain under consideration is N . Then all N grid-points may be considered as final points of, say, 96-hours backward trajectories. System (2.64) must be solved for all N trajectories in order to calculate the values of the concentrations at time t_0 at all grid-points. After that the same procedure must be repeated for $t_1, t_2, \dots, t_k$. Some details are omitted here in order to simplify the explanation of the main idea (for example, a special treatment will be necessary when some trajectories leave the space domain of the model).

The number of trajectories used in the advanced application of a Lagrangian method can be very huge. This number is $(k+1)N$, where N is normally of order 10^3 , while k is of order $4.3 \cdot 10^3$ when the time-interval is one year and it is assumed that the trajectories arrived at the grid-points after every time-interval of two hours. This means that the system (2.64) has to be solved several million times. The total number of tasks is very impressive. However, it should immediately be noted that it is not very difficult to treat them numerically. There are two reasons for such a statement:

- (i) the separate tasks can easily be treated by using fast and cheap numerical methods

and (what is even more important)

- (ii) the separate tasks are quite independent of each other and well-balanced (this means that the computational work at any fixed time t_j , $j=0,1,\dots,k$, can easily be vectorized and/or parallelized).

It is clear from (ii) that new and modern high-speed computers can efficiently be used when a Lagrangian method is applied to produce the same output as that produced by an Eulerian model (high-speed computers and their applications will be discussed in Chapter 6 - Chapter 9).

2.10.4. Advantages and disadvantages of the Eulerian models and the Lagrangian models

Eulerian and Lagrangian models are often used in air pollution studies. Each of these two groups of models have both advantages and disadvantages. Very often the advantages of the one group of models are disadvantages for the other group. Therefore it is very difficult to recommend the use of models from any of the two groups for all possible situations (although in particular situations the use of models from one of the two groups may be more advantageous than the use of models of the other group). The main advantages and disadvantages of the air pollution models belonging to these two groups are briefly discussed below (under the assumptions made in § 2.10.2).

The main advantages of using an Eulerian model to study air pollution phenomena are:

- (i) In the Eulerian models one works with system (2.63), which is derived by utilizing directly the physics of the air pollution problem that is studied, while several simplifying assumptions are as a rule imposed in the Lagrangian models in order to obtain system (2.64).
- (ii) All physical processes that are involved in the long-range transport of air pollutants are taken into account when Eulerian models are used. The diffusion process is, for example, totally ignored in the transition from (2.63) to (2.64) when Lagrangian models are to be applied.
- (iii) The three-dimensional case can be treated by Eulerian models (if both a sufficiently powerful computer and a sufficiently accurate set of input data are available), while it is practically impossible to solve the task of developing three dimensional Lagrangian models unless some rather unrealistic assumptions are imposed.

However, the Eulerian air pollution models have also at least one serious disadvantage:

- (a) The Eulerian methods are normally time-consuming. The system (2.63) is much more complicated than the system (2.64). The use of this

system is good from a physical point of view, because (as mentioned above) (2.63) reflects the physical properties of the studied problem better. However, the complexity of the system (2.63) implies a requirement for more accurate numerical methods and, thus, leads to a substantial increase of the computer time needed. The problem of finding numerical methods that are both sufficiently accurate and sufficiently fast is still an open problem when Eulerian models are used, although some success has been achieved during the last years.

The main advantages of using a Lagrangian model to study the levels of concentrations and depositions of a given group of air pollutants are:

(1) The system (2.64) is much simpler than (2.63). Therefore it is much easier to implement simple and fast numerical methods in a Lagrangian model.

(2) The Lagrangian models are conceptually much simpler than the Eulerian models. Therefore it is easy (a) to modulate them, (b) to perform different experiments with them and (c) to interpret the results obtained by them.

The Lagrangian models have, unfortunately, several disadvantages. Most of the disadvantages are caused by the assumptions made in order to obtain the simplified system (2.64) from (2.63); see § 2.10.2. The main disadvantages of the Lagrangian models that are caused by the assumptions in § 2.10.2 are:

(A) One of the physical processes that plays an important role in the long-range transport of air pollutants, the diffusion process, is completely ignored when a Lagrangian model is used.

(B) It is, as already mentioned above, practically impossible to develop three-dimensional Lagrangian models unless some very unrealistic assumptions are added to the assumptions made in § 2.10.2. Some people try to develop three-dimensional models by imposing an assumption that the wind velocity vectors have the same direction and the same magnitude at all heights. This means that $u(x,y,z,t)$ and $v(x,y,z,t)$ can vary in x , y and t , but for any fixed triple (x,y,t) these quantities are constant in z . Of course, such wind fields will **never** occur in practice when the space domain is sufficiently large and the time-interval is sufficiently long. Therefore the development of such a model is controversial. While some physical processes can be described better in such a model (compared with the two-dimensional models), there is a danger that all benefits from the better description of some of the physical processes will be lost when the errors caused by the non-physical assumption that the wind velocities do not vary in the vertical direction are dominating over the remaining errors.

(C) It is not very easy to obtain approximations for the concentrations at the starting points of the trajectories. One can try to use the calculated values at the starting points, but such a choice will lead to a requirement to keep and update at every time-step huge sets of data. The new modern computers allow us to manipulate with very large sets of data. Therefore this drawback is perhaps not as serious now as it was, say, ten years ago.

(D) Problems may sometimes occur during the calculation of trajectories. The numerical procedures that are used in the calculation of trajectories are in most of the cases convergent, but certain problems could arise when some special meteorological situations take place.

(E) If a Lagrangian model is to be used to calculate the same output as an Eulerian model on **a fine spatial grid**, then the results will normally be very unreliable and in most of the situations even misleading (the trajectories will "miss" many sources). It is at least true that some actions are to be taken in an attempt to improve the performance.

The last drawback should be taken very seriously; it indicates that one must be very careful when Lagrangian models are used in connection with fine spatial grids. The application of Lagrangian models in this situation should probably be avoided. Of course, the danger of achieving poor results with Lagrangian methods, in the case where these are used on a fine spatial grid, is well known among the scientists that are working in this field. Nevertheless, the last disadvantage deserves some additional discussion. This is why the disadvantage (E) will be studied in some more detail in the next sub-section.

2.10.5. The choice of a model among the two groups

It is not easy to recommend a model from any of the two groups that will be efficient in all possible applications. An attempt to discuss the choice, by using the advantages and the disadvantages of the models from the two groups, will be carried out below.

Let us start the discussion with the Eulerian models. It is relatively easy to draw a general conclusion concerning the use of large air pollution models from this group. From the discussion in the previous sub-section, it is clear that

an Eulerian model can always be applied when the numerical methods implemented in it are sufficiently accurate and when the CPU time that will be needed to handle the model numerically on the computer available does not cause problems.

However, it must also be emphasized that it is not easy to design both sufficiently accurate and sufficiently fast numerical methods for the computer treatment of an Eulerian model. In fact, this problem (the determination of an optimal set of both sufficiently accurate

and sufficiently fast numerical methods) is not completely solved yet. Moreover, in many Eulerian models the numerical methods are perhaps sufficiently fast, but not sufficiently accurate. In any case, more efforts in this direction are absolutely necessary. Some success has been achieved during the last years.

The situation is much more complicated when Lagrangian methods are to be used to produce output results at all points of a given grid. In these methods one often simplifies the mathematical description of the problem by inserting the simplifying assumptions (i) - (iii) from § 2.10.2. Of course, these assumptions lead to some errors in the results; these errors will be called here Lagrangian errors. However, there are also errors that are caused by other factors (discretization errors, errors due to the uncertainties in input data, errors due to the uncertainties in the physical parameters and mechanisms, etc.). These errors also appear when other models (for example, Eulerian models) are used. This second group of errors will be called common errors. It is clear that the use of a Lagrangian method is justified only in the case where the common errors are both acceptable and dominating over the Lagrangian errors.

It is difficult to answer the question: when the common errors are dominating over the Lagrangian errors (and, moreover, are still acceptable)? Assume that the discretization errors are the dominating errors among the common errors. Then an attempt to answer the above question can be done by using the simple situation that is described in Fig. 2.2. There are three sources S_1 , S_2 and S_3 and one receptor R . Assume also that the wind velocities are constant in the part of the space domain where the sources S_1 , S_2 and S_3 and the receptor R are located (the direction of the wind is shown in the plots by the trajectory that arrives in R).

It is clear that only the second source S_2 is contributing to the concentrations of the appropriate air pollutants in the receptor point R in this situation; see also Fig. 2.3. Thus, the physical background of this situation is very clear.

Assume now that the part of the space domain where the sources and the receptor are located is discretized by using a coarse (for example, 150 km x 150 km) spatial grid; such a grid is used in the EMEP models (see Eliassen and Saltbones [154], Iversen et al. [252-254] or Simpson [413-414]). The coarse grid is drawn in Fig. 2.4. The usual assumption is made that the emissions from the three sources are instantaneously spread in the grid-squares where the sources are located.

It is seen from Fig. 2.4 that both source S_2 and source S_3 will contribute to the appropriate concentrations in the receptor R , because the trajectory arriving in the receptor point R is passing through the grid-squares where these two sources are located. Thus, source S_3 will contribute to the concentrations in the receptor according to any Lagrangian model, although there is no physical justification for such a contribution.

However, it should be noted that the Eulerian models also have problems in this situation. It is easy to see that if any Eulerian model is used, then all three sources S_1 , S_2 and S_3 will contribute to the appropriate concentrations in the receptor R . Thus, both source S_1 and source S_3 will contribute to the concentrations in the receptor point according to any Eulerian model, although there is no physical justification for such contributions (i.e. the

results from the Lagrangian model will in some sense be better in this particular case).

Let us reiterate now the assumption that the discretization errors are dominating over the other errors. If the accuracy requirements are not high and if the discretization errors caused by using the coarse grid are still acceptable for the particular process studied, then both Lagrangian and Eulerian models can successfully be used.

The discussion in the previous sub-section shows that the use of coarse spatial grids in the discretization of the air pollution models, both Lagrangian and Eulerian, may cause, in some receptor points, contributions from sources for which one can easily show (by physical reasons) that these sources should not have any influence on the receptor points under consideration. Therefore it is often desirable to try to improve the performance of the models by introducing finer spatial grids. The situation where a finer grid, a (50 km x 50 km) grid, is specified is shown in Fig. 2.5. It should be mentioned here that such grids are often recommended when large air pollution models are used to study the long-range transport of air pollutants in Europe.

It can easily be deduced from Fig. 2.5 that the performance of the Eulerian models will be improved very considerably in this situation. If the diffusion process is sufficiently well described in the model, then only "the right" source S_2 will contribute to the appropriate concentrations in the receptor point R . This means that the results, which are calculated by the model when the numerical methods are sufficiently accurate, will agree with the results which should be expected when this simple particular physical situation is studied (the fact that only the second source, S_2 , will contribute to the appropriate concentrations in the receptor point R under the assumptions made above is illustrate in Fig. 2.3).

The performance of any Lagrangian model may, however, become even worse when a finer spatial grid is applied (under the assumptions made in §2.10.2). This is demonstrated in Fig. 2.5. In the particular situation considered here, the Lagrangian models will miss source S_2 (because the trajectory, see the arrow in Fig. 2.5, is not passing the grid-square where this source is located). This means that the model will report no contribution from the sources S_1 , S_2 and S_3 to the receptor R (or, in other words, the model will "lie" in this situation by telling us that the concentrations in the receptor point R are unrealistically low). The probability for missing sources will be increased dramatically if the grid is further refined; if, for example, a 10 km x 10 km grid is used.

This means that, while one should expect the Eulerian models to be in some sense "convergent" (the accuracy of the results could be improved by refining the grid⁵), the Lagrangian models do not have this useful property (at least, under the assumptions made in §2.10.2).

⁵ It must be emphasized here the some additional conditions are necessary in order to guarantee the convergence. These conditions will not be discussed here. The quotation marks are used to indicate that the definition given above is rather loose, although it seems to be quite sufficient and, what is even more important from a practical point of view, it is very useful when a model has to be chosen.

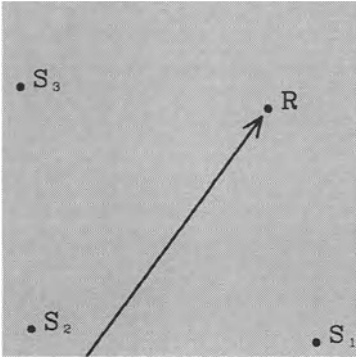


Figure 2.2
A configuration of three sources and one receptor in a part of the space domain.

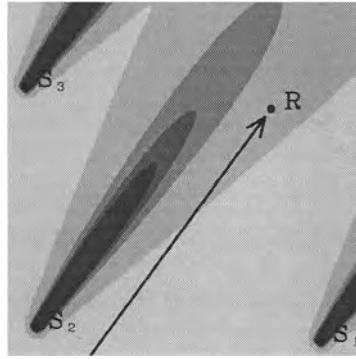


Figure 2.3
The plumes from the three sources when the wind is constant in the sub-domain

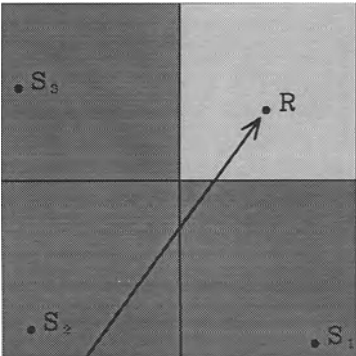


Figure 2.4
The emissions in the sub-domain in the case where a coarse grid is used.

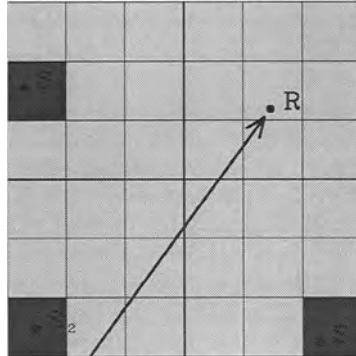


Figure 2.5
The emissions in the sub-domain in the case where a finer grid is used.

One can argue that the example shown in Fig. 2.2 - Fig. 2.5 is very stringent. Of course this is true, and in practice the situation will often not be so bad. However, the example shows clearly that one must be very careful when Lagrangian models are used to give results on fine spatial grids.

The fact that the diffusion is often completely ignored in the Lagrangian models implies very often an underestimation of the concentrations in the receptor points when finer spatial grids are used (because some sources are often missed by the trajectories). At present, many

the air pollution modellers are accepting this fact. Some of them, however, are trying to remedy the situation by introducing **compensating errors** (instead of drawing the more correct conclusion that the better choice is to avoid the Lagrangian models when the spatial grid is very fine). The idea behind the introduction of compensating errors is very simple. Assume that the modeller knows that the model tends to create errors with a certain sign (positive or negative). Then another error that has the opposite sign is introduced in order to reduce the effect of the first error⁶. This idea can be implemented in the following way for Lagrangian models on fine grids. Since sources that are relatively long away from the receptors will often be missed by the trajectories, these are simply spread. This is equivalent to introducing new artificial sources in some grid-squares (and diminishing the strength of the actual sources). It is clear that correct answers to questions as "**when to spread?**" and "**how to spread?**" are absolutely necessary when this procedure is used. Note too, that if a trajectory hits a source that has been spread, then one will achieve the opposite effect (the compensating error is not giving any compensation in this case; it will increase the total error).

In the situation shown in Fig. 2.5 the introduction of **compensating errors** means, among other things, that artificial sources in some neighbouring grid-squares (of the grid-square where source S_2 is located) will in fact be created, and the strength of the source S_2 will be reduced. The desired effect in the receptor point R could be achieved if the source is correctly spread (the appropriate concentrations in the receptor point R will be increased). On the other hand, the depositions and the chemical reactions in some parts of the region are carried out under the assumption that contributions due to artificial emission sources are added to the concentrations. Thus, one accepts that the intermediate concentrations calculated when the trajectory is followed are not correct; the hope is that the concentration in the end-point is more accurate (because an attempt has been made to take into account contributions due to emission sources close to the trajectory). The above discussion shows that the introduction of **compensating errors** is a controversial decision, and explains why such an approach should perhaps be avoided in modern models. Note too, that if one decides to spread equally the emissions in the nine grid-squares that form a large 150 km x 150 km grid-square, then one comes back to the situation described in Fig. 2.4. This means that the refining of the spatial grid will have no practical effect in this particular situation or, in other words, the refining of the spatial grid will only result in some waste of CPU-time and storage).

The introduction of **compensating errors** could be considered as one of the last remnants of the romantic era of the air pollution modelling (the 70's), when the scientists had much poorer knowledge than now about the physical and chemical processes in the atmosphere and much poorer means than now (numerical methods and, especially, computers), but attacked heroically the problems with the slogan: "**The faith moves mountains**". Now we both have much better knowledge (concerning the relevant physical and chemical processes in the atmosphere) and much better means (modern numerical methods and, what is even more important, high speed computers with enormous memory capabilities) that could successfully

⁶ From this short description it becomes immediately clear that this is not a scientific approach. There is some similarity between this approach and the methods used by a small shop-keeper in an attempt both to retain some profit and to sell quickly some products that have been a long time in the shop.

be used to build up, not heroically, but in a systematic and patient way, robust and reliable mathematical models without making crude non-physical assumptions. It should also be stressed, however, that the development of such a model is still a very difficult task and that there are still many unresolved problems and, therefore, **compensating errors** will also be used in the development of models in the near future.

Another interpretation of the above implementation of compensating errors in Lagrangian models is to consider this procedure as an attempt to introduce a kind of diffusion in the models. Instead of applying the diffusion mechanism to spread the concentrations, one applies it to spread the emissions. Thus, one is working (under the computations along the trajectory) with emissions that may differ considerably from the actual emissions in the grid-squares passed by the trajectory. The hope is that the concentrations in the end-points of trajectories are improved when such a procedure is correctly implemented.

The discussion in this sub-section indicates that one should be careful when Lagrangian models are used. This does not mean that Lagrangian models should be avoided. In some situations the use of a Lagrangian model may be a good choice or even the best choice. The main recommendations concerning the use of Lagrangian models can be summarized as follows:

1.	The situations where the simple use of a Lagrangian model can be made (see § 2.10.2) are the best example where the use of a Lagrangian model is a good choice.
2.	If the accuracy required is low and the discretization is not very fine, then the Lagrangian models will give quite acceptable results.
3.	If the spatial discretization is fine or if three dimensional cases are to be studied, then one should consider the possibility of switching to an Eulerian model.

The main conclusion is that one must be careful when a Lagrangian approach is chosen. However, many important practical tasks can successfully be solved by Lagrangian models. In some situations the Lagrangian approach may be the best choice.

One final remark is needed. **The procedure is clearly not mass conserving when a Lagrangian model is used to produce the same output as an Eulerian model.** Indeed, the deposition process is carried out along the trajectories, but the deposited mass is not taken into account; only the deposited mass at the final trajectory points is used (together with some kind of interpolation) when the total deposition in the area of consideration is to be formed. Therefore it is hard to see how the total deposition calculated in this way can be connected with the total emission in the area (taking into account also the transport over the boundaries of the space domain of the model). The violation of the conservation law (when the total deposition in the area is to be calculated) is perhaps not very important when the grid is

coarse (because the dominating errors are mainly caused by truncation errors in this situation). However, the violation of the conservation law may cause serious problems when a fine grid is in use.

2.11. Concluding remarks and references concerning large models for studying long-range transport of air pollutants

In some cases (but not always) the mathematical description of the main physical processes that take place under the long-range transport of air pollutants is, in principle at least, a straight-forward and well-understood procedure. The difficulties appear when the mathematical expressions are to be treated numerically and when some specific values are to be assigned to the physical parameters that are used in the mathematical terms. For some of these parameters (as, for example, the wind velocities) it is relatively easy to get sufficiently accurate approximations by using measurements. However, it is rather difficult to evaluate some of the other physical parameters. These problems have been discussed in **Section 2.1**. The importance of the models in the solution of several practical tasks is also discussed in Section 2.1.

The mathematical models must be reliable. The reliability of the mathematical models is normally checked by comparing the concentrations and the depositions calculated by the models with the corresponding measurements. Therefore it is not a great surprise that there are comparisons of model results with measurements in nearly all papers dealing with large mathematical air pollution models; see, for example, [22], [36], [39], [43], [60], [153-154], [215], [216], [228], [230-232], [251-254], [301], [302], [313], [314], [355], [375-376], [411-414], [463]. Many other examples can be given. The need of checking the reliability of the mathematical models is briefly discussed in **Section 2.2**.

The mathematical description of the main physical processes is given (separately for each of these processes) in **Section 2.3 - Section 2.8**. After that, a general description of the mathematical model to be studied in the following chapters is presented in **Section 2.9**. The mathematical model (2.63) is certainly not the most general model that can be constructed. Moreover, different modifications must be inserted if the model is to be used for some special purposes. However, the model described by (2.63) contains all main properties that a general-purpose LRTAP model must have. Therefore a careful study of the numerical treatment of (2.63) will be useful in all cases; even in the many cases where the formulation is not the same as that given in Section 2.9.

The direct use of the system of partial differential equations (2.63) in the numerical treatment leads to the so-called Eulerian or grid models. Another approach, the Lagrangian approach, is based on a preliminary simplification of the system of partial differential equations (2.63) to a system of ordinary differential equations (2.64), which is considered along trajectories. The models based on this approach are called Lagrangian or trajectory models. The advantages and the disadvantages of the Eulerian and Lagrangian models are discussed in **Section 2.10**. Lagrangian models are often used in practice; see, for example, [36], [101], [153-154], [215], [251-254], [335], [413-414]. These methods were rather

popular in the 70's when the computers were too slow to deal with advanced Eulerian models. Now the Eulerian models become gradually more and more popular, but the Lagrangian models can be used very successfully in many situations.

There is one case, however, where the use of Lagrangian models should perhaps be avoided; see § 210.5. This is the case where a fine grid is needed. The use of fine grids is only justified by the necessity to improve the quality of the calculated results. Thus, the basic idea is to improve the model by the introduction of fine grid. However, the improvements must be carried out in a harmonic way. The modeller must be aware that in many cases one will not improve the model if **only one** of the following conditions is satisfied:

- (i) the grid is refined,
- (ii) the quality of the input data (meteorological data and/or emission data) is improved,
- (iii) the physical mechanisms used are more accurate,
- (iv) the set of the chemical reactions are more adequate,
- (v) the numerical methods are both faster and more accurate,
- (vi) the initial and boundary conditions are treated in a more reliable way.

Very often improvements will not be achieved even if several of these conditions (but not all of them) are satisfied. One should carefully investigate all these conditions in order to decide how to satisfy this set of conditions in an optimal way. The physical processes involved in the model may be treated in a more accurate way or in a less accurate way (depending of the accuracy that is required). However, the physical laws on which these processes are based should not be violated (this is, at least, highly desirable).

CHAPTER 3

NUMERICAL TREATMENT OF LARGE AIR POLLUTION MODELS

"Paul Dirac¹ once said that it was more important to have beauty in one's equations than to have them agree well with experiment. He went on to explain that the empirical discrepancy might be due to unimportant factors which better calculations or more accurate observation would put right. The implication was that nothing will remedy the defect of mathematical ugliness".

John Polkinghorne: *"The Queen of the Sciences"*, Bulletin of the Institute of Mathematics and Its Applications, **21**(1985), 121-122.

It has been shown in Chapter 2 that any LRTAP model (any long-range transport of air pollution model) can be described mathematically by a system of partial differential equations (PDE's). The number of equations in this system is equal to the number of species which are to be studied by the model. Thus, including more species leads to an increase of the size of the problem. However, even more important is the fact that the system of PDE's cannot be solved analytically (even if the number of species is small; say two) unless some very unrealistic assumptions are imposed. Therefore the systems of PDE's, by which the large air pollution models are described mathematically, have to be treated numerically.

Many different numerical methods have been used in the treatment of large air pollution models. The number of numerical methods that have been suggested for use in large LRTAP models in different books and papers, but which have not actually been used in any production model until now, is even greater. This explains why it is not possible to describe all these methods in this book. Therefore several numerical methods that can be used to get approximate solutions of the systems of PDE's will be studied in detail in this chapter. The other possibilities will be briefly discussed. However, references, in which adequate descriptions of this second group of numerical methods can be found, will be given.

As mentioned in the previous two chapters the system of PDE's, by which a large air pollution model is described, can be split into different sub-systems. It is natural to split this system into parts (sub-systems) that correspond to the physical processes that are involved in the model. It is also natural to select (or, at least, to try to select) as good as possible numerical methods for each of these sub-systems. For every sub-model (or, in other words, for every sub-system of the system of PDE's), an attempt to describe one or two of the best numerical methods, among the numerical schemes that have been used (or have been proposed for use) in air pollution modelling until now, will be made. However, it must be reiterated here that other numerical methods may also perform rather well; references concerning many other numerical methods will be given.

¹ **Paul Adrien Maurice Dirac:** A British physicist, the winner of the Nobel prize in physics in 1933 (together with E. Schrödinger).

3.1. Splitting of the large model to sub-models

Consider the system of PDE's that describes an LRTAP model. This system, which has been presented in Section 2.9, is written in the following way:

$$\begin{aligned}
 (3.1) \quad \partial c_s / \partial t = & - \partial (u c_s) / \partial x - \partial (v c_s) / \partial y - \partial (w c_s) / \partial z & (\text{advection}) \\
 & + \partial (K_x \partial c_s / \partial x) / \partial x + \partial (K_y \partial c_s / \partial y) / \partial y + \partial (K_z \partial c_s / \partial z) / \partial z & (\text{diffusion}) \\
 & - (k_{1s} + k_{2s}) c_s(x, y, z, t) & (\text{deposition}) \\
 & + E_s(x, y, z, t) & (\text{emission}) \\
 & + Q_s(c_1, c_2, \dots, c_q) , & (\text{chemistry})
 \end{aligned}$$

where

$$(3.2) \quad s = 1, 2, \dots, q,$$

and

$$(3.3) \quad x \in [0, X], \quad y \in [0, Y], \quad z \in [0, Z], \quad t \in [0, T].$$

It is highly desirable to apply a common numerical method directly to the whole system of PDE's (3.1). Unfortunately, this is a very difficult task. The different terms in the right-hand sides of the q equations in (3.1) originate from different physical processes and, therefore, have different properties. The requirements, which have to be satisfied by the numerical methods that either have to be selected or have to be developed, depend strongly on properties of the mathematical terms on the right-hand side of (3.1). Very often, the requirements that are imposed by different terms of (3.1) work in different directions. Therefore, it is very difficult to satisfy all the requirements simultaneously. This is why some kind of splitting is to be applied (and, in fact, splitting algorithms are used in **all** known big air pollution models). Different splitting procedures have been proposed and/or discussed in the literature; see, for example, Marchuk [307], and McRae et al. [315]. The first paper, in which a splitting procedure has been proposed (although in a very simple example), is probably that written by Bagrinowskii and Godunov [20]).

The discussion in the previous paragraph indicates that it is natural to apply splitting according to the different physical processes that are involved in the model. Such a procedure has been proposed in Marchuk [307] in connection with a model that is somewhat simpler than that described by (3.1), but the main idea is fairly general. It can be applied also in connection with the system (3.1) or even in connection with systems of PDE's arising

from more complicated air pollution models. The same idea can, of course be applied also in connection with mathematical models arising in some other fields of science and engineering.

If the model is three-dimensional, then it is convenient to carry out first the computations for every horizontal plane and then to perform the modifications in the vertical direction.

The implementation of the above two ideas:

(i) splitting the air pollution model according to the physical processes involved in it

and

(ii) processing first the horizontal planes and after that carrying out computations along the vertical direction

leads to a splitting procedure that can be defined as follows:

(A) Computations in the horizontal planes. Let some z_0 be fixed in the interval $[0, Z]$. When the model described by (3.1) is discretized, the set of points (x, y, z_0) , where $x \in [0, X]$ and $y \in [0, Y]$, defines a horizontal plane within the space domain of the model. Then the computations connected with this horizontal plane (i.e. for $z=z_0$) are carried out by treating numerically the following four systems:

$$(3.4) \quad \partial c_s^{(1)} / \partial t = - \partial (u c_s^{(1)}) / \partial x - \partial (v c_s^{(1)}) / \partial y,$$

$$(3.5) \quad \partial c_s^{(2)} / \partial t = + \partial (K_x \partial c_s^{(2)} / \partial x) / \partial x + \partial (K_y \partial c_s^{(2)} / \partial y) / \partial y,$$

$$(3.6) \quad d c_s^{(3)} / dt = - (k_{1s} + k_{2s}) c_s^{(3)}(x, y, z_0, t),$$

$$(3.7) \quad d c_s^{(4)} / dt = + E_s(x, y, z_0, t) + Q_s(c_1^{(4)}, c_2^{(4)}, \dots, c_q^{(4)}) ,$$

where $s=1,2,\dots,q$ and q is again the number of pollutants that are to be studied by the model. ■

(B) Computations along the vertical lines. Assume that (x_0, y_0) is a fixed point with $x_0 \in [0, X]$ and $y_0 \in [0, Y]$. When the model is discretized, the set of points (x_0, y_0, z) , where $z \in [0, Z]$, defines a vertical grid-line within the space domain of the model. Then the computations for $(x, y)=(x_0, y_0)$ can be performed by treating numerically the following system of PDE's:

$$(3.8) \quad \partial c_s^{(5)} / \partial t = - \partial (w c_s^{(5)}) / \partial z + \partial (K_z \partial c_s^{(5)} / \partial z) / \partial z,$$

where again $s=1,2,\dots,q$ and q is the number of pollutants that are to be studied by the model. It should be noted here that the dry deposition may be incorporated in the boundary conditions of the vertical transport. ■

The above discussion shows that the system of PDE's (3.1) can be split into five PDE systems, which can (after the splitting) be solved separately at every time step. It must be mentioned here that, while the separation of the computations in the vertical direction from the computations in the horizontal planes is in many cases convenient, it is not always necessary to separate these computations. The terms in the right-hand side of (3.8) can be added to the previous systems: the first term to the system (3.4), the second one to system (3.5). In some situations this could be the better choice. Nevertheless, this situation will not be discussed in detail here, because there is nothing new when the latter choice is made: systems (3.4) and (3.5) can be handled in the same way when the extra terms from (3.8) are added. Formally, the problem becomes slightly easier (because four PDE systems, instead of five, are to be handled). On the other side, the addition of extra terms in (3.4) and (3.5) may cause some difficulties when these systems of PDE's are handled numerically; see also Section 3.8.

Of course, the five systems (3.4)-(3.8) must be coupled in some way. The process of coupling the systems that arise after the splitting of the model can be carried out, at every time-step, in the following way. At each time-step the five systems are successively solved. Assume that some approximations to the concentrations c_s at time t have been found. Then the first system is solved by using the approximations of c_s as starting values. The solution found is used as a starting vector in the treatment of the second system. This process is continued: the solution of the third system is used as a starting vector in the treatment of the fourth system and the solution of the fourth system is used as a starting vector in the treatment of the fifth system. Finally, the approximated solution of the fifth system is accepted as an approximation to the solution of the global model at time $t + \Delta t$ (where Δt is the time-stepsize) and, thus, everything is ready to start a new time-step.

Sometimes it is recommended to change the order in which the five sub-systems are solved after every time-step (see, for example, McRae et al. [315]). Thus, if $c_s^{(1)}$, $c_s^{(2)}$, $c_s^{(3)}$, $c_s^{(4)}$ and $c_s^{(5)}$ have been determined (in this order) at time-step i , the approximations at time-step $i+1$ will be calculated in a reverse order; i.e. the order at time-step $i+1$ will be $c_s^{(5)}$, $c_s^{(4)}$, $c_s^{(3)}$, $c_s^{(2)}$ and $c_s^{(1)}$.

The splitting device defined above has several advantages. The most important advantages are:

- (i) One works with simpler systems.
- (ii) Different numerical methods can be used in the solution of the different systems (and, thus, the computational work can be easily optimized).

(iii) One can prepare separate and easily exchangeable modules for the different parts of the model.

(iv) If the model is considered as a two-dimensional model, then the vertical coupling procedure defined by (3.8) should simply be omitted.

However there are also some disadvantages. For example, it is rather difficult to control the errors that are caused by the splitting procedure chosen and this is a serious drawback.

The fact that there are disadvantages should be taken into account both when splitting techniques are chosen and during their implementation in the model under consideration. It is important to emphasize here that

it is not sufficient to choose numerical methods that perform well when these are used separately.

One should also try to check in some way the global performance, i.e. whether the methods that are selected give good results also when these are combined to treat numerically the mathematical model (3.1) by some splitting technique. Appropriate sets of test-examples can be used to check the reliability of the splitting procedure applied in the model.

3.2. Treatment of the horizontal advection

It has already been mentioned in the previous chapter that it is quite sufficient to consider a single PDE in order to describe the use of different numerical methods that can be applied in connection with the horizontal advection (or, in other words, in connection with the air pollution transport that is caused by the horizontal wind velocities u and v); see Section 2.1. This means that the following single partial differential equation could be studied (and will be studied in this section):

$$(3.9) \quad \partial c / \partial t = - \partial (uc) / \partial x - \partial (vc) / \partial y,$$

instead of the system of PDE's (3.4).

In general, the horizontal wind velocities u and v depend both on the space coordinates x and y and on the time t ; i.e. $u=u(x,y,t)$ and $v=v(x,y,t)$.

There are many numerical methods which could be applied in the solution of (3.9). Most of these methods can be divided into four groups:

- (i) methods based on the use of **characteristics**,
 - (ii) methods based on the use of **finite differences**,
 - (iii) methods based on the use of **finite elements**
- and
- (iv) methods based on the use of **spectral expansions**.

Methods of the first group have been the first numerical methods that have been used in connection with the solution of equations of type (3.9). The numerical methods based on the use of characteristics are most convenient (and, thus, have been mainly used) when the coefficients u and v satisfy some special conditions. If (3.9) is to be used in connection with large air pollution problems, then u and v are rather general functions of x , y and t . This is why methods from the first group have not been very popular among the scientists working with large LRTAP models, but despite of this they have been used (and are used) in many applications; see, for example, van Loon [293] and the references there. The methods based on the use of characteristics are studied in many text-books on PDE's; see, for example, Richtmyer and Morton [365] or Smith [423]. These methods are often called semi-Lagrangian methods by the scientists.

The methods based on finite differences have systematically been used from the beginning of the 20'th century; not only in the case where (3.9) is used in air pollution modelling, but first and foremost to study other physical processes. Some basic theoretical results have been formulated in 1928 by Courant, Friedrichs and Lewy; see [76]. Methods of the second group have been commonly used in the numerical treatment of air pollution models in the 60'ies and in the 70'ies. Some of the classical finite difference schemes are not very popular when large and very large LRTAP models are to be handled. This is especially true for the finite difference schemes of first and second orders. However, more accurate finite difference schemes, and first and foremost, finite difference schemes of orders three and four, are still often used. Several comparisons of different numerical methods that can be used to treat (3.9) numerically, when this PDE is obtained by using some splitting devices applied to large air pollution models, have shown that the methods based on second order finite differences perform often worse than methods of the next two groups; see, for example, Chock [67,68], Chock and Dunker [69], Long and Pepper [292], Orszag [336-338]. Methods of this group have thoroughly been studied in all text-books on numerical solution of PDE's; see again Richtmyer and Morton [365] or Smith [423].

The methods based on the application of finite elements deserve some special attention. These methods have several very nice properties. The most important among these properties is perhaps the fact that the discretization can easily be carried out by using non-equidistant grids. This allows the modeller to use finer grids in parts of the region that are of interest or should be studied more carefully. Until now mainly simple finite elements schemes (first-order linear one-dimensional finite elements) have been used in large air pollution models. Some properties of one such scheme, which has been used by Chock [67,68], Chock and Dunker [69] and Pepper et al. [349], will briefly be discussed in this section. The results of many experiments that are reported in the literature seem to indicate

that even such simple linear one-dimensional finite element schemes work very satisfactorily in many situations. However, it is also highly desirable to try to implement more advanced (but also more complicated) finite element schemes that have the ability to perform local refinement of the grid when such a refinement has to be applied. The application of two-dimensional finite element schemes in the numerical treatment of (3.9) has been analyzed in Neta and Williams [327]; see also Neta [326].

The methods based on spectral expansions are not very well known among the modellers (although a fine description of some of the main ideas, and also implementation issues, could be found in Kreiss and Oliger [270] or Orszag [336-338]). There are two ways of using spectral expansions. The classical way is to use some spectral expansion to transform the equations involved in the model and to work with the transformed equations. The methods based on this approach are called **spectral**. It is more common, however, to apply the second way of using spectral expansions. If the second way is used, then the spectral expansions are only used to obtain some polynomials that interpolate the unknown functions (then these polynomials are differentiated in order to obtain approximations to the derivatives of the unknown functions under consideration). The method is called **pseudospectral**. It is crucial to emphasize here that the pseudospectral methods are usually based on the use of truncated Fourier expansions in order to obtain approximations to the spatial derivatives at the grid-points. The methods are also called Fourier methods when truncated Fourier expansions are used to obtain approximations to the spatial derivatives at the grid-points. Pseudospectral methods that are based on the use of truncated expansions in Chebyshev polynomials have also been proposed and studied; see, for example, Bayliss et al. [24]. In both cases (when the pseudospectral methods are based on truncated Fourier expansions and when the pseudospectral methods are based on truncated expansions in Chebyshev polynomials), trigonometric polynomials are used to interpolate the spatial derivatives of the unknown functions (in our particular case, the concentrations of the air pollutants), while algebraic polynomials are used in the other numerical methods. The pseudospectral methods based on Fourier expansions require periodicity on the boundaries. This may cause some problems, but there are devices that can be applied to deal with this requirement; see, for example, Lyness [300], Roache [367,368]. The requirement for periodicity can be avoided when Chebyshev polynomials are used, but the methods based on the use of the Chebyshev polynomials are more expensive (with regard to computing time) and, moreover, some requirements on the distribution of the grid-points are imposed by the latter methods.

All numerical methods discussed above have both some advantages and some disadvantages. It is not possible to recommend a numerical method that is the best one for all applications. In the case where (3.8) is derived from an LRTAP model, methods based on the application of finite elements as well as pseudospectral methods seem to be among the best (but, let us reiterate once again, good results can also be obtained by some other numerical methods). Some particular methods from these two groups will shortly be discussed in this section.

3.2.1. Pseudospectral discretizations

A pseudospectral discretization algorithm can be introduced in connection with the advection part of a large air pollution model in the following way. Let us consider (3.9). Assume that this partial differential equation can be rewritten in the form given by (2.7). In other words, let us consider the equation

$$(3.10) \quad \partial c / \partial t = - u \partial c / \partial x - v \partial c / \partial y,$$

$$x \in [0, 2\pi], \quad y \in [0, 2\pi], \quad t \in [0, T],$$

instead of (3.9).

Assume also that an initial value condition (i.e. the set of values of the concentrations at $t=0$) is given by

$$(3.11) \quad c(x, y, 0) = p(x, y) \quad \text{for } \forall x \in [0, 2\pi] \wedge \forall y \in [0, 2\pi].$$

Finally, assume that the boundary conditions of transport problem studied are periodic, i.e.

$$(3.12) \quad c(0, y, t) = c(2\pi, y, t) \quad \text{for } \forall y \in [0, 2\pi] \wedge \forall t \in [0, T],$$

$$(3.13) \quad c(x, 0, t) = c(x, 2\pi, t) \quad \text{for } \forall x \in [0, 2\pi] \wedge \forall t \in [0, T].$$

Introduce two equidistant spatial grids $\mathbf{X}_M$ and $\mathbf{Y}_N$ as well as a matrix $G(t)$, t being fixed in the interval $[0, T]$, as follows:

$$(3.14) \quad \mathbf{X}_M = \{ x_m \mid x_0=0, x_{m+1}=x_m+\Delta x, \Delta x=2\pi/(2M+1), m=0(1)2M \},$$

$$(3.15) \quad \mathbf{Y}_N = \{ y_n \mid y_0=0, y_{n+1}=y_n+\Delta y, \Delta y=2\pi/(2N+1), n=0(1)2N \},$$

$$(3.16) \quad G(t) = \begin{vmatrix} c(x_0, y_0, t) & c(x_1, y_0, t) & \dots & c(x_{2M}, y_0, t) \\ c(x_0, y_1, t) & c(x_1, y_1, t) & \dots & c(x_{2M}, y_1, t) \\ \vdots & \vdots & \ddots & \vdots \\ c(x_0, y_{2N}, t) & c(x_1, y_{2N}, t) & \dots & c(x_{2M}, y_{2N}, t) \end{vmatrix}$$

The trigonometric polynomials that are given below can be constructed by the use of the grids $\mathbf{X}_M$ and $\mathbf{Y}_N$ and the elements of matrix $G(t)$; see, for example, Dahlquist and Björck [93]:

$$(3.17) \quad T_{M,n} = A_n(t) + \sum_{k=1}^M [a_{k,n}(t) \cos(kx) + b_{k,n}(t) \sin(kx)]$$

$$n = 0, 1, \dots, 2N,$$

$$(3.18) \quad T_{N,m}^* = A_m^*(t) + \sum_{k=1}^N [a_{k,m}^*(t) \cos(ky) + b_{k,m}^*(t) \sin(ky)]$$

$$m = 0, 1, \dots, 2M.$$

The coefficients of these trigonometric polynomials can be calculated by using, for $\forall x_m \in \mathbf{X}_M \wedge \forall y_m \in \mathbf{Y}_N$, the following formulae:

$$(3.19) \quad A_n(t) = (2M+1)^{-1} \sum_{m=0}^{2M} c(x_m, y_n, t),$$

$$n=0, 1, \dots, 2N,$$

$$(3.20) \quad a_{k,n}(t) = 2(2M+1)^{-1} \sum_{m=0}^{2M} c(x_m, y_n, t) \cos(kx_m),$$

$$n=0, 1, \dots, 2N,$$

$$(3.21) \quad b_{k,n}(t) = 2(2M+1)^{-1} \sum_{m=0}^{2M} c(x_m, y_n, t) \sin(kx_m),$$

$$n=0, 1, \dots, 2N,$$

$$(3.22) \quad A_m^*(t) = (2N+1)^{-1} \sum_{n=0}^{2N} c(x_m, y_n, t),$$

$$m=0, 1, \dots, 2M,$$

$$(3.23) \quad a_{k,m}^*(t) = 2(2N+1)^{-1} \sum_{n=0}^{2N} c(x_m, y_n, t) \cos(ky_n),$$

$$m=0, 1, \dots, 2M,$$

$$(3.24) \quad b_{k,m}^*(t) = 2(2N+1)^{-1} \sum_{n=0}^{2N} c(x_m, y_n, t) \sin(ky_n),$$

$$m=0, 1, \dots, 2M,$$

Let r_i be the vector formed by the elements of the i 'th row of the matrix $G(t)$ and c_j be the vector formed by the elements of the j 'th column of the matrix $G(t)$. These vectors can be written as

$$(3.25) \quad r_i = \{ c(x_0, y_{i-1}, t), c(x_1, y_{i-1}, t), \dots, c(x_{2M}, y_{i-1}, t) \},$$

$$i=1, 2, \dots, 2N+1,$$

$$(3.26) \quad c_j = \{ c(x_{j-1}, y_0, t), c(x_{j-1}, y_1, t), \dots, c(x_{j-1}, y_{2N}, t) \},$$

$$j=1, 2, \dots, 2M+1,$$

It is clear now that some approximations r_i^* and c_j^* to the vectors

$$(3.27) \quad r_i^* = \{ \partial c(x_0, y_{i-1}, t) / \partial x, \partial c(x_1, y_{i-1}, t) / \partial x, \dots, \partial c(x_{2M}, y_{i-1}, t) / \partial x \},$$

$$i=1, 2, \dots, 2N+1,$$

$$(3.28) \quad c_j^* = \{ \partial c(x_{j-1}, y_0, t) / \partial y, \partial c(x_{j-1}, y_1, t) / \partial y, \dots, \partial c(x_{j-1}, y_{2N}, t) / \partial y \},$$

$$j=1, 2, \dots, 2M+1,$$

can be found by the use of (i) the elements of matrix $G(t)$, (ii) the relationships (3.17)-(3.26) and (iii) the derivatives of the trigonometric polynomials (3.17)-(3.18). Some FFT (fast Fourier transforms; see, for example, Brigham [38], Cooley and Tukey [75]) algorithm must be applied to increase the speed of computations that are carried out by using the formulae (3.17)-(3.28) (see more details in Fornberg [169], Kreiss and Oliger [270], Orszag [336-338], Zlatev [512,515,517,520,522], Zlatev and Berkowicz [526], and Zlatev et al. [529-532]).

It is easily seen that there exist two matrices, a $(2M+1) \times (2M+1)$ matrix S_M and a $(2N+1) \times (2N+1)$ matrix S_N , such that

$$(3.29) \quad r_i^* = S_M r_i, \quad i=1, 2, \dots, 2M+1,$$

and

$$(3.30) \quad c_j^* = S_N c_j \quad j=1, 2, \dots, 2N+1.$$

Let

$$(3.31) \quad g(t) = \{ r_1, r_2, \dots, r_{2M+1} \}$$

and

$$(3.32) \quad g^*(t) = \{ c_1, c_2, \dots, c_{2N+1} \}.$$

Since r_i^* and c_j^* are approximations (on the points of the grid $X_M \times Y_N$) of the derivatives $\partial c / \partial x$ and $\partial c / \partial y$, it is obvious that the partial differential equation (3.9) can be approximated (again on the points of the grid $X_M \times Y_N$) by a linear system of ordinary differential equations (ODE's):

$$(3.33) \quad dg/dt = (US + VPS^*P) g,$$

where

(i) g is an abbreviation of $g(t)$,

(ii) S and S^* are block diagonal matrices (S contains $2N+1$ diagonal blocks equal to matrix S_M , while S^* contains $2M+1$ diagonal blocks equal to matrix S_N),

(iii) U and V are diagonal matrices, the diagonal elements of which are values of functions u and v on the points of grid $X_M \times Y_N$.

It is also necessary to explain the use of the matrix P in (3.33). Observe that the vector $g(t)$ is formed by the elements of the matrix $G(t)$ when these are ordered by rows, while the vector $g^*(t)$ is formed by the elements of the same matrix when these are ordered by columns. This shows that there exists a permutation matrix P , such that

$$(3.34) \quad g^*(t) = P g(t) \quad \wedge \quad P^{-1} = P.$$

Let $z^*(t) = S^* g^*(t)$. From (3.34) it follows that the vector

$$(3.35) \quad z(t) = P^{-1} z^*(t) = P S^* g^*(t)$$

can be used to get correspondence with the coordinates of the vector $S g(t)$. Matrix P is used in the two relations (3.34) and (3.35) on which the derivation of (3.33) is based.

The method used in the transformation of the PDE (3.1) into the system of ODE's (3.33) is, as mentioned before, well-known under the names pseudospectral or Fourier method; see, for example, Fornberg [169], Gottlieb and Orszag [211], Kreiss and Oliger [270] or Orszag [336-338]. This method is suitable in many different application areas. It has been used in connection with large air pollution models by Christensen and Prahm [71], Prahm and Christensen [355], Zlatev [512,515,517,520,522], Zlatev et al. [529-535]. The derivation of the method given here follows the description of the pseudospectral algorithm in Zlatev et al. [529]. Several remarks are necessary in connection with the assumptions made during the derivation of the pseudospectral discretization of (3.1).

Remark 3.1. The assumption $x \in [0, 2\pi]$ is not a restriction. If x belongs to an arbitrary interval $[a, b]$, then this interval can be transformed to $[0, 2\pi]$ by the substitution $\xi = 2\pi(x-a)/(b-a)$. The same is true for the interval of variable y . ■

Remark 3.2. If the equation

$$(3.36) \quad \partial c / \partial t = - \partial (uc) / \partial x - \partial (vc) / \partial y + E,$$

where $E = E(x, y, t)$ is an emission source term, is solved instead of (3.9), then an extra term (a vector whose elements are the values of the emissions E on the points of the grid $\mathbf{X}_M \mathbf{x} \mathbf{Y}_N$) has to be added to the right-hand side of the system of ODE's (3.33). ■

Remark 3.3. Denote $\mathbf{R}^* = (\mathbf{US} + \mathbf{VPS}^* \mathbf{P})\mathbf{g}$. In the actual computations with the pseudospectral (or Fourier) method, one normally uses the vector $\mathbf{R}^*$. Thus, the matrices $\mathbf{S}$, $\mathbf{P}$ and $\mathbf{S}^*$ are neither calculated explicitly nor stored in the solution process. However, these matrices are needed in the stability analysis of the numerical methods for solving (3.33). ■

Remark 3.4. The pseudospectral method (based on truncated Fourier expansions) was derived for the weaker form (2.7) of the two-dimensional advection equation. It makes the analysis much simpler. It is also possible (in principle, at least) to derive the pseudospectral method for the advection equation (3.9). ■

The next step is the treatment of the system of ODE's (3.33). Systems of the same type also appear when other space discretizations are in use (such as, for example, finite differences and finite elements). However, the fact that truncated Fourier expansions are used in the space discretization imposes some specific requirements on the time-integration schemes. Therefore, it is worthwhile to discuss the problem of selecting time-integration schemes that perform well for pseudospectral methods. The choice of a time-integration scheme is crucial when an attempt is made to reduce the computational work (time-integration schemes that can be applied with large time-stepsizes lead to a reduction of the time-steps needed and, thus, to an efficient treatment of the model). This is why the discussion of the time-integration schemes will be postponed for Chapter 5, where efficient implementations of some numerical methods will be studied.

3.2.2. Finite element discretization

Consider now (3.9) instead of the weaker form (2.7). Many discretization schemes which are based on the implementation of different finite elements can be applied during the numerical treatment of the spatial derivatives in this equation. It has been mentioned that one-dimensional linear finite elements are rather popular when large air pollution models are to be treated. This is one of the simplest discretization methods. However, the truth is that in many practical applications it may be desirable to apply more accurate methods (based on two-dimensional finite elements).

It should be emphasized here that the application of any finite element scheme (that is based on two-dimensional finite elements) leads, after the discretization of the right-hand-side of (3.9), to a system of ODE's of the following type:

$$(3.37) \quad P dg/dt = Hg + f,$$

$$P \in \mathbf{R}^{N \times N}, \quad H \in \mathbf{R}^{N \times N},$$

$$dg/dt \in \mathbf{R}^N, \quad g \in \mathbf{R}^N, \quad f \in \mathbf{R}^N;$$

where the notation used can be explained as follows:

- (i) N is the number of grid-points used in the space discretization,
- (ii) P is a symmetric and positive definite matrix which depends only of the finite elements chosen,
- (iii) H is a matrix that depends on the wind velocities (and, therefore, the matrix H is in general time-dependent),
- (iv) $g = g(t)$ is a vector that contains the values of the concentrations at the grid-points and at time t ,
- (v) $dg/dt = dg(t)/dt$ is a vector that contains the first time derivative of the concentrations stored in vector $g(t)$,
- (vi) $f = f(t)$ is a vector, whose components depend on the boundary condition used in the definition of the problem (for the sake of simplicity, it will be assumed here that Dirichlet boundary conditions are always specified; this means that the values of the concentration, which is studied by (3.9), are given on the boundaries).

Let us assume that the space domain is a square whose sides are parallel to the coordinate axes, and that equidistant grid lines with $\Delta x = \Delta y$ are used. The number of grid-points on a grid-line parallel to the Ox axis is equal to the number of grid-points on a grid-line parallel to the Oy axis when this assumptions are made. Let us denote the number of grid-points in a grid-line by n (it is clear that $n^2 = N$). Then both P and H are banded matrices with the same structure of the non-zero elements and with bandwidth equal to $2n+1$ or to some number that is close to $2n+1$. Moreover, only a few diagonals of P and H contain non-zero elements. These facts about the structure of the non-zero elements of the matrices P and H must be used in order to achieve an efficient treatment of the model under consideration.

It should be noted here that the application of any finite difference scheme will lead to an ODE system of the same type as (3.37), but with $P = I$, where I is the identity matrix. Thus, the use of finite difference schemes leads in general to computations that are in some sense simpler than the computations with finite elements. It should be noted here that the lumped mass principle is sometimes used to get $P = I$ also when a finite elements discretization is used. However, these devices are not very popular. Devices that are based

on the use of the lumped mass principle will lead to some reduction of the computations; the price that has to be paid is some loss of accuracy.

The system (3.37) has to be treated numerically. Most of the methods for solving systems of ODE's are designed for problems of the type: $dg/dt = h$, where h is a vector. Therefore it is desirable to transform the system (3.37) into a system $dg/dt = h$. Formally this can be done by multiplying both sides of (3.37) by P^{-1} . The resulting system is

$$(3.38) \quad dg/dt = P^{-1}(Hg + f),$$

Let us denote by h the right-hand-side vector in (3.38). This means that vector h is given by

$$(3.39) \quad h = P^{-1}(Hg + f).$$

One should find an efficient method for calculating the vector h . This vector must be computed once or several times at every time-step. There are three major approaches that can be applied in the calculation of the vector h .

(A) Calculating the inverse of matrix P . It seems to be worthwhile to invert matrix P in the very beginning and then to use it again and again; every time when vector h is to be calculated. Such a procedure can be applied because P is a constant matrix. However, this will in general be inefficient because, while P is a banded matrix, its inverse P^{-1} will in general be a dense matrix. The inversion of the matrix P is more expensive than the factorization of the same matrix and, therefore, it is not recommended to calculate P^{-1} (see the further discussion).

The calculation of the inverse of a matrix is a linear algebra operation and, therefore, standard FORTRAN subroutines are available in many libraries; as, for example, the NAG Library [325] or LINPACK (see Dongarra et al [112]). The subroutines from LAPACK (Anderson et al. [8]) are becoming more and more popular. It is rather easy to attach any of these subroutines to the advection modules of large air pollution models. ■

(B) Solving (3.39) by a direct method. In the very beginning of the computations, one can also factorize matrix P into a product of some easily invertible matrices (as, for example, triangular and/or orthogonal matrices). This can be done by several different methods, but the simple Cholesky decomposition is often the most efficient algorithm (because matrix P is symmetric and positive definite). If the Cholesky decomposition is used, then an upper triangular matrix L is obtained, such that $P = L^T L$ (if the rounding errors caused by the use of computer arithmetic are neglected). Since the matrix P is a banded matrix, the upper triangular matrix L is also a banded matrix whose bandwidth is equal to the upper bandwidth of P (to one half of the bandwidth of P , because P is symmetric).

The Cholesky decomposition is an important and often used linear algebra operation. Standard FORTRAN subroutines for performing the Cholesky decomposition can be found in the same places where standard subroutines for calculating the inverse of a matrix can be found; i.e. in [8], [112], [325]. Also such a subroutine can easily be attached to the advection part of a large air pollution model. ■

(C) Using some iterative method. A third possibility is to apply some iterative method. Let us assume that the conjugate gradient method is to be used. The convergence of this iterative methods is ensured because P is symmetric and positive definite, but may be very slow. The most time-consuming operation is a multiplication Pz where z is a vector.

Iterative methods for solving systems of linear algebraic equations are discussed in many papers and books on numerical linear algebra. The books of Varga [456] and Young [495] are classical; many iterative methods are studied in [10], [14]-[19], [23], [44], [116], [121], [147], [148], [160]-[164], [175], [224], [236], [257], [261], [265], [266], [310], [316], [318], [321], [339], [340], [362]-[366], [377]-[380], [421]-[423], [430], [466], [478], [480], [493]-[499], [522]-[524]. Standard FORTRAN subroutines are available in many libraries; such as the NAG Library [325]. There exist also well-documented and carefully tested packages that contain many subroutines using different iterative methods. ITPACK ([265], [266], [498], [499]) and PCGPACK (which is based on methods and procedures developed at Yale University, the methods used in this package being described in [147], [148], [377]-[379]) are two such packages. ■

Let us try now to evaluate the number of simple arithmetic operations needed in the calculation of vector h from (3.39) by the three methods sketched above. If the first approach is used (the approach based on a direct use of the inverse matrix P^{-1}), then the number of operations needed to calculate vector h is

$$(3.40) \quad \text{COST}_1 = O(N^2) .$$

The number of simple arithmetic operations needed with the second approach (where the matrix L obtained by the Cholesky decomposition is used to solve the system directly) is

$$(3.41) \quad \text{COST}_2 = O(2Nn) .$$

This means that the computations with P^{-1} will be about $n/2$ times more expensive than the computations needed when the second approach is used.

The number of simple arithmetic operations needed with the third approach, where the conjugate gradient algorithm, or some other iterative method is used to solve the system (3.39), is

$$(3.42) \quad \text{COST}_3 = O(Njp),$$

where j is the number of diagonals that contain non-zero elements (this number is small, typically 5, 7 or 9) and p is the average number of iterations per calculation of h (averaged over the total number of vectors h that are needed during the whole computational process).

It is difficult to compare COST_2 with COST_3 . It should be stressed here that the constant in the right-hand-side of (3.42) is considerably larger than that in (3.41), because of the overhead due to the stopping criteria that have to be used during the iterative process. The quantity p in (3.42) is dependent both on the particular iterative method that is selected and on the particular problem that is to be solved. This quantity is not known in advance (before the start of the computations), as all the other quantities that are used in (3.40) and in (3.41). This means that the optimal solution should be found by comparing the results from several trials with different iterative methods for the problem that is to be solved with the results obtained by the second approach.

The disadvantages of the use of iterative methods were pointed out in the previous paragraph. However, these methods have also several advantages. The most important advantage of the iterative methods is perhaps the fact that the iterative methods seem to be more suitable for most of the modern high-speed computers than the direct methods.

It has been mentioned several times (see also the beginning of this sub-section) that the use of one-dimensional linear finite elements is rather popular when air pollution models are to be treated numerically. Assume that the space domain is a rectangle with sides parallel to the coordinate axes (the last part of this assumption is not a restriction, because if the sides of the rectangle are not parallel to the coordinate axes, then these could be made parallel to the coordinate axes by a simple transformation of the coordinate system).

Consider now the first-order space derivative $\partial(uv)/\partial x$. This derivative can be discretized on each of the grid-lines parallel to the Ox axis by using one-dimensional linear finite elements. In this way a system of ODE's of the same type as (3.37) will be found for each grid-line parallel to the Ox axis. The treatment of $\partial(vu)/\partial y$ is quite similar. Also here the use of one-dimensional linear finite elements will lead to one system of ODE's of the same type as (3.37) for each grid-line parallel to the Oy axis.

Let us assume that the number of grid-points along the grid-lines parallel to the Ox axis is m , while the number of grid-points along the grid-lines parallel to the Oy axis is n . Then the application of any one-dimensional linear finite elements scheme will lead to $m+n$ systems of ODE's of the same type as (3.37). However, the systems are much smaller; the number of equations in every system is either m or n , while the number of equations in (3.37) is $N=mn$. Even more important is the fact that the matrices involved in any of the $m+n$ systems are tri-diagonal. The exploitation of this fact leads to efficient numerical algorithms. It is relatively easy to design schemes that will perform efficiently on high-speed computers. These are the advantages of the scheme. However, these advantages have been achieved by using computational schemes that are of low order of accuracy (first-order of accuracy). Therefore the problem of achieving the required accuracy may be a difficult task

when these methods are used.

Assume that the vector h has been computed in some way. Then it still remains to solve the system of ODE's $dg/dt = h$ (or $m+n$ such systems of ODE's when one-dimensional linear finite elements are used). Some time-integration scheme has to be applied in the solution of this system (these systems) of ODE's. The problems arising in the numerical integration of the system (or systems) $dg/dt = h$ are similar to those arising after the pseudospectral discretization of the first order derivatives by which the advection process is described mathematically (see in the end of the previous sub-section. A common description of the choice of a time-integration scheme for systems of ODE's arising in different parts of air pollution models will be given in Chapter 5.

3.3. Algorithms for the horizontal diffusion terms

The horizontal diffusion terms in the air pollution models can be treated in a similar way as the advection terms were treated in the previous section. Therefore it is not necessary to go into details; a short discussion of the numerical algorithms that can be applied in connection with the horizontal diffusion sub-model (3.5) is quite sufficient.

It is often assumed that the horizontal diffusion coefficients are non-negative constants. In some models it is assumed that these coefficients are zero (i.e. there is no horizontal diffusion). This is perhaps an extreme point of view. In this section it will be assumed that the diffusion coefficients are positive constants (but sufficiently small, so that the advection dominates over the diffusion in the model). If such an assumption is made, then the system of PDE's (3.5) is reduced to the following system of PDE's:

$$(3.43) \quad \partial c_s^{(2)} / \partial t = K_x \partial^2 c_s^{(2)} / \partial x^2 + K_y \partial^2 c_s^{(2)} / \partial y^2,$$

$$s=1, 2, \dots, q.$$

As mentioned before, everything concerning the numerical treatment of the diffusion part of the model can be described by considering one PDE only. This means that it is quite sufficient to consider the equation

$$(3.44) \quad \partial c / \partial t = K_x \partial^2 c / \partial x^2 + K_y \partial^2 c / \partial y^2.$$

The numerical treatment of the diffusion equation (3.44) is very similar to the numerical treatment of the advection equation (3.9). Therefore only the major differences are briefly discussed below.

(A) Use of finite elements or finite differences. Assume that either finite differences or finite elements are to be used in the discretization of the spatial derivatives. Then (3.44) will be reduced to a system of ODE's which is of the same type as the system (3.37) (where P is again the identity matrix when finite differences are used). However, the matrix H will now be a constant matrix that is symmetric and positive definite. This might be exploited during the numerical treatment.

A straight-forward way of handling numerically the horizontal diffusion is to add the horizontal diffusion terms to the horizontal advection terms (i.e. to treat the horizontal advection and the horizontal diffusion together). This can easily be done when methods based on finite differences or finite elements are used. The advantage of combining the advection terms with the diffusion terms is that only one system of PDE's is to be solved when this is done; instead of the two systems (3.9) and (3.44). The disadvantage is (or, at least, may be) the fact that matrix H is now formed by contributions both from the elements that are used to discretize the first order spatial derivatives in (3.9) and from elements that are used to discretize the second order derivatives in (3.44). Thus, the matrix H is no longer symmetric and positive definite as in the case where (3.44) is to be solved. Moreover, the matrix H has no longer the properties of the matrix with the same name that appears when (3.9) is solved. This means that the methods discussed in Sub-section 3.2, or at least some of them, may be inefficient. It seems to be worthwhile to check carefully their efficiency. The conclusion is that the coupling of the advection and the diffusion terms is possible when finite elements or finite differences are used and it may be worthwhile to do this, but a careful check of the performance of the numerical algorithms in the new situation is needed. Efficient algorithms for the matrix H that arise from the coupled advection and diffusion terms must be found and this may be a difficult task.

If one decides to treat the diffusion independently from the advection, then it will in general be much easier to find efficient methods that exploit the fact that matrix H is symmetric and positive definite. However, one must solve two systems of ODE's at every time-step when the advection and the diffusion are treated separately. Only one system is to be solved in the case where the advection and the diffusion terms are combined. The question whether it is better to solve two systems with two different and very efficient methods or to solve only one system, but with a method which is more complicated and perhaps not very efficient, seems to be open. Much more experiments (than these available in the literature until now) are needed in order to give an adequate answer to this question. ■

(B) Use of a pseudospectral discretization. The analysis given above shows clearly that both finite differences and finite elements can be applied directly to (3.44) or the diffusion terms and the advection terms can be combined and both finite differences and finite elements can be applied to the combined systems. The implementation technique is in principle the same as for the case where (3.9) is solved (see the previous section), however there may be difficulties with the choice of the time-integration scheme. The situation is practically the same when a pseudospectral technique is to be used. One can use the trigonometric polynomials (3.17) and (3.18) in order to get some approximations to the second order derivatives $\partial^2 c / \partial x^2$ and $\partial^2 c / \partial y^2$ at the grid-points; these polynomials must now be differentiated twice (not only once as in the case where only the advection is treated; see the previous section). Everything is quite straightforward and there is no need to go into further details here.

The fact that the horizontal advection phenomena are dominating over the horizontal diffusion phenomena, when long-range transport of air pollution is studied, can be exploited to design some special algorithms for the treatment of the horizontal diffusion. One of these special algorithms is sketched in Zlatev et al. [533] and used in **the Danish Eulerian Model** (which has been studied in [525]-[528], [537]-[540]). This algorithm is in fact semi-analytical; the assumption that the horizontal diffusion coefficients are positive constants is also exploited in the derivation of this special algorithm. The special algorithm is based on using the representation of the solution by a Fourier expansion. This expansion is truncated and the truncated sum is used as an approximate solution. It is convenient to truncate the Fourier expansion by keeping as many terms as the number of grid-points that are used along the grid-lines parallel to the O_x and O_y axes. More details about this algorithm can be found in [533]. ■

(C) Use of some methods of characteristics. If methods based on characteristics are used in the treatment of the advection terms, then the diffusion terms are to be treated with some other method (finite elements, finite differences or some of the pseudospectral discretizations). ■

3.4. The chemical terms and the problem of stiffness

"When the equations represent the behaviour of a system containing a number of fast and slow reactions, a forward integration of these equations becomes difficult".

H. H. Robertson: *"The solution of a set of reaction rate equations"*. In: **"NUMERICAL ANALYSIS: AN INTRODUCTION"** (J. Walsh, ed.), pp. 178-182. Academic Press, New York-London, 1966.

The chemical part of a large air pollution model is nearly always the most time-consuming part. Therefore it is very important to handle this part by applying efficient numerical methods. Unfortunately, it is very difficult to design numerical methods by which the system of ODE's arising from the chemical part of the model can be treated sufficiently fast and, moreover, by which sufficiently accurate results can be obtained.

The numerical methods that can be applied during the treatment of the chemical part of a large LRTAP model may be divided into three major groups:

- (i) numerical methods in which an attempt is made to exploit some specific features of the problem being solved (some features of the chemical reactions),
- (ii) classical numerical methods for solving stiff systems of ODE's,
- (iii) numerical methods based on some extrapolation techniques.

The methods of the first group are often based on heuristics. It is difficult to obtain some error estimation when these methods are used. However, if such methods are used with sufficiently small time-steps, then they will as a rule produce sufficiently accurate results. It should also be noted that the computational work per time-step is rather low for these methods. However, it may be difficult to determine precisely what is "a sufficiently small time-step"? The situation is rather confusing: while everybody seems to accept the fact that the methods of this group are to be run with a sufficiently small time-step, the time-steps that are actually used in different air pollution models vary in a very wide range.

The methods of the second group are based on a solid theoretical basis. It is possible to obtain some error estimation when some of these methods are used. The computational work per time-step for the classical numerical methods for stiff ODE's is normally larger than that for the methods from the first group. The computational work per time-step increases rather quickly when more accurate methods are to be used (this is especially true for high-order methods of Runge-Kutta type). However, it is often possible to apply the more accurate methods with a larger time-step (it should be noted that it seems to be necessary to try to keep the errors under control when the methods are used with large time-steps).

There are many theoretical results concerning the numerical methods of the third group. An error estimation can be calculated in a natural way when numerical methods based on extrapolation rules are used. These methods are normally more expensive than the methods from the other two groups. It is again possible to run them with larger time-steps.

It is clear (from the short discussion given above) that the methods of any of the three major groups have both some advantages and some drawbacks. This fact shows why it is difficult to recommend the methods of only one of these groups. Some more information is needed when a good choice has to be made. Therefore some representative numerical methods of each group will be discussed in some more detail below.

3.4.1. Methods based on exploiting special features of the chemical reactions

The so-called quasi-steady-state-algorithm is a typical representative of these methods. A good discussion of this method has been given by Hestvedt et al. [235], in 1978. However, the method has been well-known (and it has often been used; not only in air pollution problems, but also in many other scientific and engineering problems) a long time before 1978; see, for example, Chameides and Stedman [63], Dimitriades et al. [111], Farrow and Edelson [165] or Jeffries and Saeger [258].

The quasi-steady-state-algorithm can shortly be described as follows. Consider only the system of ODE's arising from the chemical reactions:

$$(3.45) \quad \frac{dc_s}{dt} = Q_s(c_1, c_2, \dots, c_q), \quad s = 1, 2, \dots, q.$$

The s 'th equation of this system of ODE's can also be written in the following way:

$$(3.46) \quad dc_s/dt = P_s - L_s c_s,$$

where both $P_s = P_s(c_1, c_2, \dots, c_q)$ and $L_s = L_s(c_1, c_2, \dots, c_q)$ are in general non-linear terms, which are respectively called the production term and the loss term for the s 'th pollutant. Moreover, the relationships $P_s \geq 0$ and $L_s \geq 0$ are satisfied for all species and for all values of $t \in [0, T]$.

Assume that some time-stepsize Δt is to be used at some time-step n . **The quasi-steady-state-algorithm is based on the assumption that sufficiently accurate results could be achieved by treating each equation in system (3.46) as an independent equation.** For each equation one studies the magnitude of the quantity $\Delta t L_s$ and, according to the magnitude of this quantity, one applies one of the following three formulae:

$$(3.47) \quad c_s^{n+1} = P_s/L_s,$$

$$\text{for } \Delta t L_s > 10,$$

$$(3.48) \quad c_s^{n+1} = P_s/L_s + (c_s^n - P_s/L_s) \exp(-\Delta t L_s),$$

$$\text{for } 0.01 < \Delta t L_s \leq 10,$$

$$(3.49) \quad c_s^{n+1} = c_s^n + \Delta t (P_s - L_s c_s^n),$$

$$\text{for } \Delta t L_s \leq 0.01,$$

Formula (3.47) can formally be obtained from (3.46) by setting dc_s/dt equal to zero. Thus, a **steady state** case is assumed when $\Delta t L_s > 10$.

Formula (3.48) is the exact solution of (3.46) under the assumption that both the production term P_s and the loss term L_s are constants in some small neighbourhood of the interval $[t_n, t_{n+1}]$, where $t_{n+1} = t_n + \Delta t$. Note that under this assumption the equation (3.46), which is in general non-linear, reduces to a linear ODE with constant coefficients.

Formula (3.49) is an application of the simplest numerical method for solving systems of ODE's; i.e. the first-order forward Euler algorithm. A thorough description of this method is given, for example, in Henrici [233]. One should expect this algorithm to be stable when the condition $\Delta t L_s \leq 0.01$ is satisfied.

The method could be simplified and/or improved in several ways. The number of choices can be reduced from three to two if (3.48) is also used when $\Delta t L_s \leq 0.01$. More accurate integration methods (of order two or three) can be applied instead of the first-order forward Euler algorithm (3.49). However, it must be pointed out here that the main drawbacks of the methods from this group:

(i) the assumption that the equations of the system of ODE's can be decoupled so that each of them can be treated independently from the others

and

(ii) the assumption that P_s and L_s can be considered as constants when $0.01 < \Delta t L_s \leq 10$

also remain when such improvements are performed. An attempt to avoid the errors caused by these two assumptions (or at least to reduce them) can be made by introducing an iterative procedure, in which the values of P_s and L_s are updated at each iteration. However, it is difficult to prove that such an iterative method will converge: In fact, it is rather easy to prove (by counter-examples) that such an iterative procedure will in general not converge. In most of the applications of these methods (at least, in the cases where the chemical part of large air pollution models are treated by such algorithms) one performs only a fixed number of iterations (say, five) without any attempt to check whether the iterative procedure converges or not and without any attempt to evaluate the accuracy of the solution obtained after the performance of the prescribed number of iterations. This means that the iterative process is carried out according to the following scheme (NFIX being the fixed number of iterations that are to be carried out, while NSPECIES = q):

```

DO ITERS=1,NFIX
  DO S=1,NSPECIES
    Calculate an approximation
    to the value of  $c_s^{n+1}$ .
  END DO
  DO S=1,NSPECIES
    Update the values of
     $P_s$  and  $L_s$ .
  END DO
END DO

```

Very often the iterative process is further simplified by assuming that sufficiently accurate concentrations of some of the species can be calculated during the first iteration. Assume that the number of these species is $q_1 < q$. Then the next iterations are carried out only for the remaining $q_2 = q - q_1$ species.

The methods of this group are based on several assumptions which seem, at the first look at least, to be rather unrealistic. On the other hand, however, the methods have been extensively tested on many rather hard problems. The experiments indicate that the methods may give reliable and sufficiently accurate results when the time-stepsize is sufficiently small. In Hesstvedt et al. [235] a time-stepsize $\Delta t = 30$ sec. is recommended. This recommendation is based on experimental results; many experiments have been performed by choosing some typical situations (both with various typical starting concentrations and with an attempt to simulate diurnal variations of the concentrations that are involved in the chemical scheme). The recommendation to carry out the computations with a time-step not larger than $\Delta t = 30$ sec. seems to be good for the species that are involved in many of the large air pollution models. Unfortunately, such a small time-stepsize, $\Delta t = 30$ sec., is too small for practical runs of large air pollution models on long time-intervals (say, one month). Therefore, it is not a great surprise that these methods are very often used with much larger time-stepsize in many large air pollution codes. A time-stepsize $\Delta t = 900$ sec. is used in, for example, Simpson [413-414]. Many experiments were carried out at the National Environmental Research Institute by chemical schemes that are used in large air pollution models. The results of these experiments seem to confirm the statement originally made in Hesstvedt et al. [235]: the recommended value of the time-stepsize should be as stated there ($\Delta t = 30$ sec.) and, moreover, the use of a larger stepsize might lead (in some situations, at least) to rather inaccurate results. Some interesting new ideas concerning the quasi-steady state algorithm are discussed in Verwer and van Loon [462]. Note that the algorithm is called pseudo-steady-state-algorithm in [462].

3.4.2. Classical time-integration methods

The classical time-integration method are well described in the literature. However, there arise many difficulties when classical time-integration methods are to be applied in the chemical part of large air pollution problems. It has already been emphasized in Chapter 2 that the systems of ODE's which appear in the numerical treatment of the chemical parts of large air pollution models are often both very stiff and extremely badly scaled. Moreover, a rather quick and very large changes of the concentrations take place when the so-called photo-chemical reactions are switched on and off (normally at sunrise and sunset respectively). Therefore special integration methods for solving stiff systems of ODE's seem to be necessary (these are described in all textbooks on numerical solution of ODE systems; see, for example, Butcher [55], Gear [182], Hairer and Wanner [227], van der Houwen [245], Lambert [280,281] and Stetter [434]). Both the fact that the systems are badly scaled and the fact that rapid changes do occur in certain periods impose (typically during sunrises and sunsets) an additional requirement for accurate numerical methods (i.e. numerical methods of order higher than one may be needed).

Let us first consider the numerical treatment of the problem of stiffness. Some notation is needed in order both to simplify (3.45) and to achieve an agreement with the notation which is commonly used when the problem of stiffness is handled numerically in the ODE papers and books (see the above references). Denote

$$(3.50) \quad \mathbf{y} = (c_1, c_2, \dots, c_q)^T$$

and

$$(3.51) \quad \mathbf{f}(\mathbf{y}) = (Q_1, Q_2, \dots, Q_q)^T$$

By using the notation that has been introduced in (3.50) and (3.51), the system of ODE's (3.45) can be rewritten as

$$(3.52) \quad d\mathbf{y}/dt = \mathbf{f}(\mathbf{y}).$$

The integration methods for stiff systems of ODE's are necessarily implicit methods (see Butcher [55], Dahlquist [87], Hairer and Wanner [227] or some of the other references given in this sub-section). Assume that the simplest implicit integration method, the backward Euler algorithm, is used. The application of the backward Euler algorithm to the system of ODE's given by (3.52) leads to the following step-by-step integration algorithm for obtaining approximations $\mathbf{y}_n$ to the vector $\mathbf{y}(t_n)$ (applied successively for $n=1,2,\dots,N$ and $t_n = t_{n-1} + \Delta t$):

$$(3.53) \quad \mathbf{y}_n = \mathbf{y}_{n-1} + \Delta t \mathbf{f}(\mathbf{y}_n).$$

It is straight-forward to apply the functional (or the simple, or the fixed-point) iteration to obtain an approximation to $\mathbf{y}_n$ at time-step n . One simply begins with some starting approximation $\mathbf{y}_n^0$ (the upper index shows the number of the current iteration, while the lower one shows the number of the current time-step) and carries out an iterative procedure by using

$$(3.54) \quad \mathbf{y}_n^{i+1} = \mathbf{y}_{n-1}^i + \Delta t \mathbf{f}(\mathbf{y}_n^i), \quad i=0, 1, 2, \dots,$$

until some stopping criterion is satisfied. Unfortunately, this simple procedure does not work in practice; it converges only if the time-stepsize Δt is unrealistically small (see, for example, Fatunla [166] or Lambert [280,281]). It must be emphasized here that this conclusion is also true for any other implicit integration method (not only for the backward Euler algorithm).

The above brief discussion explains why any implicit integration method leads to the solution of one or several systems of algebraic equations at each time-integration step. If the system of ODE's is linear, then linear systems of algebraic equations appear, and these could be solved by direct methods (such as those that are based on Gaussian elimination). If

the system of ODE's is non-linear, then the systems of algebraic equations are also non-linear. This means that iterative methods have necessarily to be used. It should be emphasized here that the system of ODE's given by (3.52) is nearly always non-linear when this system arises from the chemical part of large air pollution models.

The systems of algebraic equations are usually solved either by the classical Newton method or by some quasi-Newton method (the name modified Newton method is also used). In any of these two cases several systems of linear algebraic equations are to be solved. The coefficient matrix of these systems contains the Jacobian matrix $J(y) = \partial f / \partial y$ of the right-hand-side vector $f(y)$ in (3.52).

More precisely, the following type of systems are to be solved at each time-step when the classical Newton method is used in the solution of (3.52) by any implicit integration algorithm:

$$(3.55) \quad (I - \Delta t \beta J(y_n^i)) \Delta y_n^{i+1} = -y_n^i + \Delta t \beta f(y_n^i) + b_n,$$

where

(i) the lower index shows the number of the time-step,

(ii) the upper index shows the number of the iteration,

$$(iii) \Delta y_n^{i+1} = y_n^{i+1} - y_n^i,$$

(iv) β is a parameter that depends on the time-integration algorithm chosen

and

(v) vector b_n depends on the approximate values of y that has been calculated at previous time-steps (i.e. this quantity does not vary during the iterations at time-step n).

The coefficient matrix $I - \Delta t \beta J(y_{n+1}^i)$ is in fact a shifted Jacobian matrix of the right-hand-side vector f of the system of ODE's (3.52). It has to be factorized at every iteration when the classical Newton method is used. This is in general an expensive operation. Therefore several modifications of the Newton method are often used in an attempt to reduce the number of computations. The first modification of the classical Newton method is based on the idea to factorize the shifted Jacobian matrix only once; in the beginning of the iterations at the time-step under consideration; i.e. to factorize the shifted Jacobian matrix $I - \Delta t \beta J(y_{n+1}^0)$. In this modification of the classical Newton method, the iterative process based on (3.55) has to be replaced by the following iterative process, where one tries to apply some factorization of the shifted Jacobian matrix $I - \Delta t \beta J(y_{n+1}^0)$ at each iteration:

$$(3.56) \quad (I - \Delta t \beta J(Y_n^0)) \Delta Y_n^{i+1} = -Y_n^i + \Delta t \beta f(Y_n^i) + b_n.$$

It can be proved that under certain assumptions both iterative processes, (3.55) and (3.56), are convergent. However, the convergence rate of (3.56) is usually slower; while the convergence rate of (3.55) is quadratic, the convergence rate of (3.56) is linear. Therefore it is necessary to check the convergence rate and to update the iteration matrix in (3.56) every time when the convergence rate is judged to be slow. Thus, it is as a rule allowed to occasionally update the factorization of the iteration matrix during the iterative process. If the time-stepsize Δt is large, then even the classical Newton method (3.55) may be slowly convergent. Therefore the convergence rate has to be checked also when (3.55) is used. If the convergence rate is very slow, then it may be worthwhile to reduce the time-stepsize and to repeat the iterative procedure, for the step where this happens, with the smaller value of Δt .

Another modification of the classical Newton methods is based on the use of a factorization of the shifted Jacobian matrix obtained at some previous time-step $j < n$ (and some iteration μ during the time-step j). Such a modification is very popular when systems of ODE's are treated numerically. The iterative process is carried out by using the following formula:

$$(3.57) \quad (I - \Delta t \beta J(Y_j^\mu)) \Delta Y_n^{i+1} = -Y_n^i + \Delta t \beta f(Y_n^i) + b_n.$$

The convergence may be slow when the last modification is used. However, in the codes for solving systems of ODE's the standard procedure is to update the shifted Jacobian matrix after a prescribed number, say *ITERS*, of iterations if the accuracy required is not achieved. Parameter *ITERS* is normally small (*ITERS*=3 is often used), but some authors recommend the use of large *ITERS* in some situations.

The classical Newton method together with the modifications described above is the most popular procedure used when systems of ODE's are to be solved numerically. However, some other promising methods have recently been suggested. Methods that are based on the use of conjugate-gradient type iterations, such as GMRES, CGS, BICGSTAB and TFQMR (usually together with some kind of preconditioning), have been used in several studies. Good results have been reported in [40]-[42], [64], [65], [189], [319]. Nevertheless, the situation is not quite clear yet. While it is obvious that such methods will work well for some kinds of problems (especially problems where the shifted Jacobian matrix is symmetric and positive definite), they will depend strongly on the preconditioners for systems of ODE's whose Jacobian matrices are general and, moreover, it may be a very difficult task to find a good and cheap preconditioner. The conclusion is that many more experiments, where preconditioned conjugate gradient type of methods are used to handle systems of ODE's arising from the chemical part of large air pollution models, are needed in order to answer the question about the efficiency of the implementation of these methods in air pollution models.

Even simpler iterative methods, such as the Jacobi iteration or the Gauss-Seidel iteration, can be used for some special systems of ODE's. However, it is well-known that these two methods do not converge or are slowly convergent (unless a very small time-step is used) for problems arising in the chemical part of large air pollution models. On the other side, the use of some block-formulations of either the Jacobi iteration or the Gauss-Seidel iteration, where the specific properties of the particular chemical reactions used in the model are taken into account to obtain dominant diagonal blocks, seem to be possible and may be efficient (i.e. it may be possible to use it with rather large time-stepsizes). Some results in this direction has recently been reported in Hertel et al. [234]. The main advantage of the method is its efficiency (with regard to the computing time used) when the method is correctly implemented on the particular computer that is to be used. However, there are also certain disadvantages. For each particular chemical scheme one has to find the right block-partitioning of the Jacobian matrix. Once such a block-partitioning is obtained, the algorithm will probably be efficient in many important, from the practical point of view, situations. If several new chemical reactions are to be added to the chemical scheme or if the chemical scheme is to be modified in some way (for example, by changing the rate constants of some chemical reactions), then a new partitioning may be needed (and if this is the case, then a new partitioning must be found). It is as a rule not an easy task to find a new partitioning. Even when the chemical scheme is unchanged, there may occur (although rather rarely) meteorological situations, in which the diagonal blocks are no longer dominant. Nevertheless, the experiments that have been performed in Hertel et al. [234] seem to indicate that the method works rather well in practice; at least for the condensed CBM IV scheme (the condensed CBM IV chemical scheme has been discussed in Chapter 2). It is not very clear if the methods will also perform well when some other chemical schemes are used.

It is important to emphasize that the convergence of the iterative procedure (in any of the implementations that were discussed above) is only a necessary condition for obtaining reliable results. Unfortunately, the convergence of the iterative procedure is by far not sufficient to guarantee sufficiently accurate results. Therefore the accuracy of the approximations obtained by any of the above procedures must also be checked. This is a very difficult and time-consuming task. Therefore it is often (in fact nearly always) omitted when large LRTAP models are treated numerically. It must be emphasized, however, that it might be rather dangerous to omit the error control during the time-integration of the chemical part. There are several reasons for this; the fact that normally numerical procedures of low order are used is perhaps the most important reason (very often, if not always, the lowest possible order, order one, is used in the time-integration algorithms by which the chemical parts of the air pollution models are treated).

The well-documented ODE codes, such as the codes described and/or discussed in [56], [83], [84], [105]-[109], [183], [184], [240]-[242], [271]-[276], [328], [329], [351], [352], [393]-[408], [450], [545], perform an automatic accuracy check during the numerical integration. The same is true for the ODE codes included in the standard FORTRAN libraries (as, for example, the codes in the well-known NAG Library; see [209], [325]). If the modeller will design his own time-integration method for the chemical part, then the accuracy control can be based on the evaluation of the local truncation error of the numerical method chosen. Devices by which such an evaluation can be obtained are described in many textbooks and papers in which the numerical solution of systems of ODE's is studied; see, for example, [3], [44]-[57], [61], [62], [78]-[95], [145], [146], [155]-[159], [167], [173], [174],

[182]-[191], [212], [225]-[227], [233], [249], [255], [259], [260], [262], [263], [291], [304]-[306], [330], [341], [356]-[357], [381]-[383], [415]-[420], [431]-437], [442], [447]. Sometimes it is better to perform an accuracy check based on an evaluation of the global truncation error. Different evaluations of the global truncation error have been proposed recently in the literature; see [418], [436], [437], [500], [501]. It should be stressed here that the accuracy checks based on some evaluation of the global truncation error are both more robust and more reliable than those based on an evaluation of the local truncation error. However, the codes where an evaluation of the global truncation error is implemented are also much more time-consuming.

As mentioned above, there are a lot of standard codes that are based on classical time-integration schemes. The code developed by Gear ([183], [184]) is very popular among the physicists, but this code is very old. The code LSODE, prepared by Hindmarsh [241], and based on the same algorithm (the backward differentiation formulae), is much better, but this code is not used very much in the numerical treatment of large air pollution models (not as much as the original code that has been developed by Gear in the end of sixties). It is not very clear why this is so. It may be advisable to try to develop a code especially for air pollution problems, in which the photochemical reactions are directly taken into account in the development of the code (very quick changes of the concentrations take place when the photochemical reactions are switched on and when these are switched off). However, it must be emphasized here that the development of such a code is not an easy task.

3.4.3. Methods based on extrapolation rules

The use of extrapolation methods is another possibility when the chemical part of an air pollution method is to be handled numerically. These methods are well-described in the literature. They have been introduced, for systems of ODE's, by Bulirsch and Stoer [44] and Gragg [212]. After that extrapolation methods have been extensively studied and used by Deuffhard and his co-workers; [105]-[109]. They developed also a code, LARKIN, that has been used in the treatment of chemical reactions. A good exposition of the main ideas used in the extrapolation methods could be found in Hairer et al. [226] and in Hairer and Wanner [227]; see also [263] and [442]. All extrapolation methods are based on the Richardson idea ([363], [364]), which is applicable not only in the solution of system of ODE's, but also in the solution of many other problems.

The use of extrapolation methods in the numerical treatment of the chemical part of large air pollution problems has several advantages:

- (i) it is relatively easy to develop codes based on extrapolation rules,
- (ii) high accuracy can be achieved,
- (iii) the methods are relatively robust (at least when some kind of smoothing is used),

and

(iv) an error estimate can cheaply be obtained.

However, there are also some drawbacks. Theoretically, the extrapolation methods are based on the fact that for sufficiently small values of time-stepsize Δt there exists an asymptotic expansion of the solution that could be truncated and used as an approximation of the solution. Unfortunately, the values of the time-stepsize Δt for which this procedure is valid may sometimes tend to zero in presence of stiffness; see, for example, Hairer and Wanner [227]. Moreover, the methods based on extrapolation are rather time-consuming (and this is certainly not desirable when large air pollution problems are handled numerically).

A careful comparison of the advantages and the drawbacks, and also many experiments performed with systems of ODE's that arise from chemical problems ([108]-[109]), indicate that some efficiently implemented extrapolation methods can successfully be used in large air pollution models.

Many different classical implicit integration methods can be applied as basic methods in an extrapolation procedure. Two simple time-integration algorithms, the trapezoidal rule,

$$(3.58) \quad Y_n = Y_{n-1} + 0.5 \Delta t [f(Y_{n-1}) + f(Y_n)],$$

and the implicit mid-point rule,

$$(3.59) \quad Y_n = Y_{n-1} + \Delta t [f(0.5(Y_{n-1} + Y_n))],$$

are often used. Consider a given time-step n and a given time-stepsize Δt . Assume that approximations to y_n are calculated (starting from y_{n-1}) by using the following sequence of time-stepsizes

$$(3.60) \quad \{\Delta t_1 = \Delta t, \quad \Delta t_2 = \Delta t/2, \quad \Delta t_3 = \Delta t/4, \quad \Delta t_4 = \Delta t/8, \quad \dots\}.$$

Denote by $Y_{j,l}$ the approximation to y_n obtained by a stepsize Δt_j (starting from y_{n-1}). Then a triangular array of vectors can be obtained by using the recurrent formula:

$$(3.61) \quad Y_{j,k+1} = Y_{j,k} + (Y_{j,k} - Y_{j-1,k}) / [(\Delta t_j / \Delta t_{j-k})^2 - 1],$$

where $j=2,3,\dots$ and $k=1,2,\dots,j-1$. It is clear that the first four rows of the triangular array are given by

$$\begin{array}{cccc}
 (3.62) & Y_{1,1} & & & \\
 & Y_{2,1} & Y_{2,2} & & \\
 & Y_{3,1} & Y_{3,2} & Y_{3,3} & \\
 & Y_{4,1} & Y_{4,2} & Y_{4,3} & Y_{4,4}
 \end{array}$$

An error estimation can be obtained by using

$$(3.63) \quad \text{ERROR}_j = \| Y_{j,j} - Y_{j,j-1} \| ,$$

where $\|Z\|$ is some norm of vector Z . It is clear that ERROR_j can be used to stop the extrapolation process at step n : if ERROR_j becomes less than some prescribed accuracy requirement ϵ , then one sets $y_n = Y_{j,j}$ and everything is ready to start the next step (step $n+1$). This is a nice property: one can start with a large time-stepsize Δt . The method will adaptively find the proper stepsize that gives sufficiently accurate results and stop the extrapolation when such a stepsize is obtained. However, the starting stepsize should not be very large: this may cause stability problems. A smoothing procedure is normally used together with the extrapolation methods in order to avoid, among other things, instabilities. If a smoothing procedure is to be applied, then the values of $Y_{j,1}$ are first smoothed and the triangular array (3.62) is built with the smoothed $Y_{j,1}$.

The methods based on extrapolation are very flexible. It is not necessary to use the sequence of stepsizes given in (3.60), although this is a choice that is often made. Any other sequence which is monotonically decreasing can be applied instead of that given in (3.60). The two integration methods (3.58) and (3.59) are often chosen because these lead to an expansion in powers of $(\Delta t)^2$. Other integration methods can also be used, but one should be aware that other methods could lead to an expansion in power of Δt . If this is the case all formulae given above will be still valid, but the second power in the denominator of (3.61) must be removed. If, for example, the backward Euler formula (3.53) is used, then the extrapolation procedure leads to an expansion that is linear in Δt . The following recurrence formula:

$$(3.64) \quad Y_{j,k+1} = Y_{j,k} + (Y_{j,k} - Y_{j-1,k}) / (\Delta t_j / \Delta t_{j-k} - 1) ,$$

is to be used in this situation instead of (3.61) when an extrapolation based on the backward Euler formula is applied.

There are, as mentioned in the beginning of this sub-section, several well-known codes that are based on extrapolation rules. The code LARKIN (a code for treating problems from the field of LARge chemical KINetics), see Deuflhard and Nowak [108] or Deuflhard et al. [109], has been used in many chemical applications, but not yet in the numerical treatment of the chemical schemes that appear in large LRTAP models.

3.5. Treatment of the deposition terms

After the splitting of the system of PDE's (3.1) into sub-models according to the physical processes, the deposition sub-model is represented by a system of ODE's:

$$(3.66) \quad \frac{dc_s^{(3)}}{dt} = - (k_{1s} + k_{2s}) c_s^{(3)}(x, y, z_0, t), \quad s=1, 2, \dots, q.$$

It has been assumed that the dry and wet deposition coefficients, k_{1s} and k_{2s} ($s=1,2,\dots,q$), may depend on both the first two spatial variables (the horizontal variables x and y) and on the time variable t ; i.e. $k_{1s} = k_{1s}(x, y, z_0, t)$ and $k_{2s} = k_{2s}(x, y, z_0, t)$. This means that (3.66) is a linear system of ODE's. Moreover, the equations in the system (3.66) are independent of each other. Therefore every equation could be solved exactly (and this is probably the best way of treating this part of the model when the above assumptions are made). Moreover, it is quite clear that (3.66) could be treated easily on massively parallel machines (there is no need for communication; each processor has to perform its own task).

If the model is considered as a three dimensional model, then dry deposition operations are performed only in the lowest layer of the space domain. Therefore it is convenient to include the dry deposition terms in the boundary conditions on the bottom of the space domain. If this is done, then the terms containing k_{1s} should be removed from (3.66). If the dry deposition terms are not included in the boundary conditions, then k_{1s} should be zero on all layers but the lowest one.

Consider again the three dimensional case. The wet deposition is a removal process due to the fact that air pollutants are absorbed in the rain drops. Therefore this process takes place only in the clouds and below the clouds. If some information about the layers which are below the clouds is available, then (3.66) has to be solved only in these layers.

Let s be fixed. Sometimes it might be assumed that k_{1s} and/or k_{2s} depend on the concentration c_s (as an illustration only, such an example has briefly been discussed in Section 2.5). In this way some of the equations become non-linear (and, therefore must be treated numerically). Thus, the problem becomes slightly more complicated, but it is still suitable for massively parallel machines.

The most complicated situation arises if one assumes that the dry and/or wet deposition coefficients, k_{1s} and k_{2s} ($s=1,2,\dots,q$), depend in a non-linear way on the concentrations. In this situation (3.66) can be rewritten as

$$(3.67) \quad \frac{dc_s^{(3)}}{dt} = - S(c_1, c_2, \dots, c_q) \quad s=1, 2, \dots, q.$$

This system is very similar to the system that arises in the chemical part of the model; see (3.7). Therefore it is appropriate to use methods similar to those studied in the previous section when (3.67) is to be handled numerically.

3.6. Initial value conditions

The system of PDE's, (3.1), by which a large air pollution model is described mathematically, has to be treated together with some initial value conditions. This means that it is assumed that the concentrations are given for $t=0$; i.e.

$$(3.68) \quad c_s(x, y, z, 0) = p_s(x, y, z), \quad s=1, 2, \dots, q,$$

where $p_s(x, y, z)$ are given functions.

The use of the initial values (3.68) in an air pollution problem should formally not cause any problem. However, the truth is that (3.68) does cause problems. The main problem is that very often the functions $p_s(x, y, z)$ are not known and it is difficult to find some approximations to these functions that can immediately be used to start the computations.

A practical device that is normally used by the air pollution modellers is to start the computations several days before the actual starting day. It is not very easy to decide how many days are to be used to start up the model. In most of the runs that were carried out with **the Danish Eulerian Model** (results of some of these runs are reported in Zlatev et al. [538], [539]) the computations were always started five days before the actual starting point. If, for example, the actual starting day is July 1, then the computations are started on June 26.

Some starting values are needed also when several days are used to start up the model. However, there is no need for very accurate starting values when several extra days are used to start up the model. Some values of the concentrations that are typical for the season can successfully be applied in this situation (although these are rather inaccurate).

3.7. Boundary value conditions

The system of PDE's (3.1) by which an air pollution is described mathematically has to be treated together with some boundary conditions. Let us consider first (3.1) as a two-dimensional system of PDE's. Then the boundary conditions can be written in the following general form:

$$(3.69) \quad \alpha_{s1}(y, t) c_s(0, y, t) + \beta_{s1}(y, t) \partial c_s(0, y, z) / \partial x = a_{s1}(y, t),$$

$$(3.70) \quad \alpha_{s2}(y, t) c_s(x, y, t) + \beta_{s2}(y, t) \partial c_s(x, y, z) / \partial x = a_{s2}(y, t),$$

$$(3.71) \quad \alpha_{s3}(x, t) c_s(x, 0, t) + \beta_{s3}(x, t) \partial c_s(x, 0, z) / \partial y = a_{s3}(x, t),$$

$$(3.72) \quad \alpha_{s4}(x, t) c_s(x, Y, t) + \beta_{s4}(x, t) \partial c_s(x, Y, z) / \partial Y = a_{s4}(x, t),$$

where $s = 1, 2, \dots, q$, while α_{si} , β_{si} and a_{si} ($i=1, 2, 3, 4$) are some given functions.

Dirichlet conditions are normally used on the horizontal boundary when air pollution models are to be treated numerically. Such conditions can be specified by taking $\alpha_{si} > 0$ and $\beta_{si} = 0$. It is clear that, under these assumptions, the boundary conditions can be rewritten as

$$(3.73) \quad c_s(0, Y, t) = b_{s1}(Y, t),$$

$$(3.74) \quad c_s(X, Y, t) = b_{s2}(Y, t),$$

$$(3.75) \quad c_s(x, 0, t) = b_{s3}(x, t),$$

$$(3.76) \quad c_s(x, Y, t) = b_{s4}(x, t),$$

where $b_{si} = a_{si}/\alpha_{si}$ and $s=1, 2, \dots, q$, $i=1, 2, 3, 4$. This is the simplest form of the boundary conditions: it is assumed that the concentrations are known on the boundaries. In principle at least, it is not difficult to get some approximations to the values of b_{si} (for example, values that are obtained by using a model defined on a larger space domain and treated by using a coarser grid can be used as approximations to these functions).

Some numerical methods require special conditions on the boundaries. This is, for example, the case when the pseudospectral algorithm is used: periodical boundary conditions are needed in this situation; see Section 3.2. One must be careful when such numerical methods are used in the model.

Let us consider (3.1) as a three-dimensional system of PDE's. The boundary conditions described above can also be applied for three dimensional models (these have to be used in all layers when the model is three-dimensional). However, it is necessary to specify additionally a lower boundary condition (on the bottom of the space domain) and an upper boundary condition (in the top of the space domain).

A Neumann boundary condition is often specified on the lower boundary of the space domain:

$$(3.77) \quad \partial c_s(x, y, 0, t) / \partial z = 0.$$

Nearly always, however, the dry deposition is incorporated in the lower boundary condition. It is relatively easy to achieve such an inclusion; one should only apply

$$(3.78) \quad \partial c_s(x, y, 0, t) / \partial z = b_{s5}(x, y, t),$$

instead of (3.77). In (3.78) $b_{s5}(x, y, t)$ is a function that depends on the dry deposition velocity of the s 'th pollutant. In the case where no dry deposition is assumed for a given pollutant (there are such pollutants), the boundary condition (3.78) reduces to the boundary condition (3.77).

As a rule a Dirichlet boundary condition is assumed on the upper boundary. This means that the boundary condition is given by

$$(3.79) \quad c_s(x, y, Z, t) = b_{s6}(x, y, t),$$

where $b_{s6}(x, y, t)$ is some function of the concentration of the s 'th pollutant in the free troposphere. Either discrete values of this function or some simple analytic expressions are normally determined by using measurements. However, there are only few measurements available for the concentrations of the pollutants in the free troposphere. Therefore the functions $b_{s6}(x, y, t)$ that are used in the large air pollution models are rather uncertain.

Sometimes it is necessary (for some pollutants, at least) to assume some exchange of air pollutants between the boundary layer and the free troposphere (i.e. between the space domain of the model and the space over the upper boundary). This can be achieved if the Dirichlet boundary condition is replaced by a more general boundary condition that is given by

$$(3.80) \quad \alpha_{s6}(x, y, t) c_s(x, y, Z, t) + \beta_{s6}(x, y, t) \partial c_s(x, y, Z, z) / \partial z \\ = a_{s6}(x, y, t),$$

where $\alpha_{s6}(x, y, t)$, $\beta_{s6}(x, y, t)$ and $a_{s6}(x, y, t)$ are some given functions. Such a boundary condition seems, for example, to be necessary in connection with ozone. It should be mentioned that the determination of the functions $\alpha_{s6}(x, y, t)$, $\beta_{s6}(x, y, t)$ and $a_{s6}(x, y, t)$ is in practice even more difficult than the determination of $b_{s6}(x, y, t)$ in the Dirichlet boundary condition (3.79). Therefore one should be very careful when such a condition is to be specified. One way out of the difficulties is to use a large Z (assuming that if the upper boundary is very high most of the pollutants will be equal to zero or, at least, to some small background values that could be prescribed in a sufficiently reliable manner).

It is necessary to incorporate the boundary conditions in the numerical methods that are used in the solution of the system of PDE's (3.1). In many cases this is a straightforward procedure. Sometimes, however, there may arise difficulties. This is especially the case with high order finite differences in which many grid-points are involved. In this situation one may need values of the concentrations not only on the boundaries, but also at some other grid-points that are close to the boundaries. Such cases will not be considered here. This is not necessary in the most of the cases. Indeed, if additional values of the unknown functions (the

concentrations) are required, then in the documentation of the method it is described how to obtain these additional values. Many modern and well-written codes are supplied with subroutines that automatically handle these problems when they arise.

3.8. Treatment of the vertical advection and diffusion

It has mainly been assumed that the model is a two-dimensional model in the previous sections of this chapter. Let us consider now a three-dimensional model. Then the terms involving the space derivatives with regard to z , the terms $-\partial(wc_s)/\partial z$ and $\partial(K_z \partial c_s / \partial z) / \partial z$, can either be considered together with the horizontal advection and the horizontal diffusion or as a separate sub-system of PDE's.

The first situation is not very interesting; it has shortly been discussed after (3.8). Some additional terms are to be included in (3.4) and (3.5). Some methods as those discussed in Section 3.2 and Section 3.3 can be applied. The major difficulty is caused by the fact that it is highly desirable to apply non-equidistant space-grids in the vertical direction, while equidistant space-grids are often used in the horizontal planes. One should be more careful during the implementation of the numerical methods when non-equidistant space-grids are used. Some of the methods used in the horizontal directions are not applicable when the terms $-\partial(wc_s)/\partial z$ and $\partial(K_z \partial c_s / \partial z) / \partial z$ are added to (3.4) and (3.5) and when the space-grids are not equidistant. For example, it will be difficult (and practically impossible) to apply the pseudospectral algorithm in this situation. It will also be difficult to apply finite differences. This means that the most suitable methods will be the methods based on finite elements. In principle, at least, the same ideas as those described in Sub-section 3.2.2 can also be applied in the situation studied here.

Assume now that the second approach is used; i.e. the vertical terms are separated from the other terms during the splitting procedure. Consider the system of PDE's (3.8) which describes the changes of the concentrations in the vertical direction. In fact, it is quite sufficient to consider one PDE only in this situation. By using this observation, one can replace here the system (3.8) by the equation:

$$(3.81) \quad \partial c / \partial t = - \partial (wc) / \partial z + \partial (K_z \partial c / \partial z) / \partial z,$$

where c can be any of the concentrations involved in the model. Pseudospectral algorithms cannot be applied here, because as mentioned above the grids used along the vertical grid-lines are not equidistant. The methods based on finite differences cannot be used either (or, at least, the use of such methods is not very easy). **Finite elements are applicable.** The same ideas as those discussed in Sub-section 3.2.2 can be used also when (3.81) is treated numerically by finite element discretizations.

The above equation is a one-dimensional problem that has to be solved many times. Let us reiterate that q is the number of pollutants involved in the model and denote by N_x and

N_y the numbers of grid-points along the lines parallel to the Ox and Oy axes respectively. Then the number of one-dimensional problems (3.81) that are to be solved at a given time-step is $q \cdot N_x \cdot N_y$. This number is equal to 35840 when $N_x = 32$, $N_y = 32$ and $q = 35$ and to 322560 when $N_x = 96$, $N_y = 96$ and $q = 35$. This explains why it is important to solve this problem efficiently.

3.9. Use of local refinement in the discretization process

Sometimes more detailed information both about the emissions and about the meteorological data is available in some parts of the space domain. In such a situation it is desirable to exploit better this information. This could be done if finer grid is used in the parts of the domain where more detailed information about the input data is available. One could use the finer grid on the whole space domain, but it is clear that such a procedure will be rather inefficient. **Therefore one should be able to use some local refinement of the grid when this is needed.** It should be pointed out here that in the situation described above the local refinement of the grid is **static** (the parts of the space domain, where more detailed information about the input data is available, are prescribed in advance).

Local refinement of the grid may also be necessary in parts of the space domain where there are quick changes of the concentrations (large concentration gradients). Such quick changes may be caused by powerful emission sources in some parts of the space domain and/or by quick variation in the meteorological conditions.

If the large concentration gradients are caused mainly by powerful emission sources, then the local refinement of the grid could be static (because the location of the powerful emission sources can be determined in advance). However, the large concentration gradients depend also on the meteorological conditions (for example, on the wind and the temperature) even if the meteorological conditions do not vary too quickly. Therefore, in this situation it is much better to apply some local refinement of the grid, which is **dynamic**. This could be explained as follows. Assume that a powerful emission source X is located in some part Y of the space domain. Consider only the wind. Assume that several days the winds around Y are from South. Then local refinement in the part Y and in some neighbouring parts located North from Y is needed. Let us assume now that after several days the winds change direction; say, the winds are from West. It is clear that the local refinement area should contain again the part Y but now together with some neighbouring parts located East from Y .

If the large concentration gradients are caused by some rapid changes of the meteorological conditions, then the local refinement of the grid have to be performed dynamically (in the parts of space domain where the rapid changes of the meteorological conditions take place; these parts are in general not known in advance). The case where the local refinement is to be carried out dynamically is usually much more difficult than the case where the local refinement is to be used statically (in a prescribed in advance part of the space domain).

Local refinement of the grid is a well-known procedure, which has successfully been used in many areas of engineering and science. It is not very popular in the field of air pollution modelling. Nevertheless, it has been used by some modellers; see, for example, [331], [332], [453], [454], [461]. It is necessary to start to use more modern procedures for automatically performing local refinement of the grid when such procedures are needed and/or where they are needed. A simple study of the gradients of the concentrations (performed automatically in the code) will show whether a local refinement procedure is needed or not and, moreover, **where** to refine the grid. An investigation of some error estimate (performed again automatically by the code) will show **how** to refine. This shows that some local refinement procedure can, in principle at least, be carried out automatically.

There are many well-described (in the literature) algorithms to perform this procedure automatically; see again [331], [332], [453], [454], [461]. It is necessary to adjust some of these procedures for air pollution models and, what is even more important, to investigate carefully the efficiency of various procedures and to select the best candidates among them. The task is by far not easy. It may turn out that some quite new procedures are needed for the large LRTAP models, and the development of such procedures may be very difficult. However, such procedures are highly desirable in many simulations with the new modern air pollution models. Therefore the development of both static and dynamic local refinement procedures is likely to become one of the most important tasks that are to be tackled in the near future.

3.10. Concluding remarks and references concerning the numerical treatment of large air pollution models

The mathematical description of the model is given in **Section 3.1**. In this section it is also explained why it is necessary to use splitting procedures, and some commonly used splitting procedures are described. The splitting procedures are usually based on an idea that has been proposed (in connection with air pollution models) by Marchuk [307]. Marchuk advocates to split a large air pollution model according to the physical processes involved in it. However, some other approaches, as for example these based on splitting the model to one-dimensional sub-models, which are studied in McRae et al. [315], can also be applied. All splitting procedures have both some advantages and some disadvantages; the advantages and the disadvantages are also discussed in **Section 3.1**. Splitting techniques are discussed in [20], [141], [142], [267], [308], [348], [491]. At present all large air pollution models are treated numerically by applying some splitting techniques.

The treatment of the horizontal advection is studied in **Section 3.2**. The main groups of methods used in the discretization of the advection terms are shortly described. The question:

"Which method is the best one for the horizontal advection?"

is still open. The pseudospectral method (based on the use of trigonometric interpolants that are truncated Fourier expansions) is a good choice, because it preserves rather well the high concentrations ([24], [67]-[71], [169], [172], [211], [270], [287], [289], [317], [336]-[338], [367], [368], [475], [512], [520], [529]-532]). Methods that are based on finite elements are most efficient when non-equidistant grids are to be used, but the truth is that the finite elements are mainly used on equidistant grids in the large air pollution models ([67]-[70], [292], [348], [349]). However, there are a few exceptions; see, for example, [331], [332]. The finite elements are rather efficient even when these are used on equidistant grids. This is why these two groups (pseudospectral methods and finite element methods) are studied in more detail. However, a good implementation of finite difference methods (and other related to them methods such as different corrected flux techniques) may also give good results; see, for example, [1]-[2], [9], [31]-[35], [50], [67]-[69], [73], [74], [77], [81], [141]-[143], [181], [213], [214], [217], [243], [248], [267], [285], [286], [297], [324], [350], [359], [365], [371], [374], [376], [387], [424]-[429], [440], [446], [452], [467], [502]. Other promising discretization methods (as, for example, different methods based on the use of characteristics, which are also called semi-Lagrangian methods) are proposed in [28], [98], [288], [293], [301], [302], [358], [360], [361], [369], [370], [388], [390], [433], [445], [490].

There are many comparative studies of methods for horizontal advection ([67]-[69], [213], [292], [349], [387]). However, all these comparisons are mainly carried out on a fixed spatial grid (as a rule on a 32×32 grid) and on sequential computers. New comparisons, where high-speed computers are used and where an attempt is made to determine an optimal size of the spatial grid for each of the compared methods, seems to be necessary. Such comparisons will give much more information because: (i) methods that show poor performance on sequential machines may perform better on parallel or vector machines and (ii) methods that are not accurate a coarse grid may give accurate results on a finer grid (being still competitive with regard to the CPU time needed).

If the space discretization of the horizontal advection part is successfully done, then it is not difficult to select a good time-integration algorithm (because the errors caused by the space discretization methods are normally dominating over the errors caused by the time-integration algorithms, and, thus, the choice of a time-integration algorithm is not very critical). The discussion of the time-integration algorithms for systems arising after the discretization of the horizontal diffusion terms is postponed to Chapter 5, where the time-integration algorithms will be related to some efficiency requirements connected with an adaptive choice of the time-step.

The space discretization of the horizontal diffusion terms in the system of PDE's (3.1) is discussed in **Section 3.3**. For many numerical methods it is possible to consider these terms together with the horizontal advection terms (but this cannot be done if some of the methods based on characteristics has been applied in the advection part). It is also possible to consider them separately. The advantages and the disadvantages of these two approaches are shortly discussed. The main conclusion is that it is much easier to treat numerically the horizontal diffusion than the horizontal advection in large air pollution models.

The numerical treatment of the chemical part is the most difficult task among the tasks that are to be solved when a large air pollution model is handled on a computer. It is crucial to solve this task by an efficient numerical method. If this task is not solved by a sufficiently

fast numerical methods, then it will be impossible to treat numerically the model, even if a large high-speed computer is available. The treatment of the chemical part is studied in **Section 3.4**. Three groups of numerical methods can be applied in the numerical treatment of the chemical part:

(i) methods based on exploiting special features of the chemical reactions (as, for example, the quasi-steady state algorithms that are studied in [63], [111], [165], [235], [258]),

(ii) methods based on classical integration formulae ([3], [6], [7], [21], [29], [30], [37], [72], [82]-[97], [99], [110], [118], [119], [155]-[159], [166], [182-191], [220], [221], [247], [250], [256], [259], [260], [268], [269], [278], [283], [351], [352], [357], [372], [373], [393]-[408], [415]-[420], [457], [458], [460], [470]-[472], [508]; see also the references in Sub-section 3.4.2)

and

(iii) methods based on extrapolation rules ([44], [105]-[109], [212], [225]-[227], [442]).

It is difficult to decide what will be the optimal choice. The methods of the second and the third group may be implemented with a rather reliable and robust error estimations, while the methods from the first group tend to be faster than those of the other two groups (however, some experiments performed by Deuflhard and Nowak [108] seems to indicate the fact that an efficient implementation of an extrapolation method may be competitive even with the quasi-steady state algorithms). The conclusion is that methods from the second and the third groups should in general be preferred. However, if the computing time is a very important factor (and this is very often the case), then methods of the first group may be selected.

The matrices arising in the chemical part of the air pollution models are often sparse (i.e. they contain only a few zero elements). This property may sometimes be exploited by applying special methods for sparse matrices; see, for example, [123]-[140], [149]-[152], [192]-[203], [206]-[208], [210], [222], [223], [354], [384]-[386], [409], [410], [448], [449], [451], [505], [506], [509], [510], [513], [514], [516], [519], [542]-[563].

The numerical treatment of the deposition terms is discussed in **Section 3.5**. Simple deposition terms are discussed. These terms lead (after the splitting procedure) to a system of linear ODE's in which each equation can be considered independently from the other equations. Thus, every equation of this system can easily be solved exactly. Several cases, where more complicated deposition terms are used, are briefly discussed.

The problem of finding some reliable initial value conditions for the system of PDE's, by which large air pollution models are described mathematically, is discussed in **Section 3.6**.

Boundary value conditions are needed when large air pollution models are handled numerically. The boundary conditions that might be used are discussed in **Section 3.7**.

Calculations in the vertical direction are briefly discussed in **Section 3.8**. The development of three-dimensional air pollution models leads to a very considerable increase of the computing time needed when the model is treated numerically (very often by a factor of 10-20; depending on the number of layers used in the model). Moreover, it is very difficult to satisfy the requirement for high-quality three-dimensional input data (both meteorological data and emission data). Simplifying assumptions are often used. The developers must be very careful; some of the simplifying assumptions may lead to significant errors and, thus, may counterminimize any benefit which can be achieved by the use of calculations in the vertical direction (note that as a rule more than 95% of the computational work will be a waste of both CPU time and storage when this happens).

Local refinement of the space grids might be needed in certain situations. Both static and dynamic local refinement techniques are shortly discussed in **Section 3.9**. Local refinement of the space grids is an important issue which is often underestimated or simply neglected in the field of air pollution modelling, while it is popular in some other fields of science and engineering. One should consider the application of techniques that are used in these fields or technique proposed by numerical analysts (as, for example, [331], [332], [453], [454], [461]) in large air pollution models when local refinement is needed in order to solve the problems more efficiently.

CHAPTER 4

TESTING THE RELIABILITY OF THE NUMERICAL ALGORITHMS

"A rather difficult advection calculation, in two dimensions, is the rotating cone problem".

G. McRae, W. R. Goodin and J. H. Seinfeld: *"Numerical Solution of the Atmospheric Diffusion Equation for Chemically Reacting Flows"*, J. Comput. Phys., **45**(1982), p.23.

It is highly desirable to implement sufficiently accurate numerical algorithms in an LRTAP model. One of the main requirements is to obtain a model where the errors due to the use of numerical procedures are smaller than the error caused by other factors (as, for example, the uncertainties in the input data, the uncertainties of some of the physical parameters, the uncertainties of the implementation in the model of some physical processes, etc.). If this requirement is satisfied (i.e. if the numerical algorithms are sufficiently accurate), then the numerical errors have no influence on the results and, thus, one will be able to study successfully different problems connected with the physical nature of the model (sensitivity of the results to variations of different physical parameters, sensitivity of the results to variations of some of the emission sources in the space domain, sensitivity of the results to variation of some meteorological parameters, etc.).

On the other hand, it is rather difficult to satisfy the requirement for sufficiently accurate numerical algorithms. There are two major reasons for this. The first reason is based on the fact that the use of very accurate algorithms and very fine discretizations will make the numerical treatment of the problem practically impossible (even if the powerful modern computers are in use). However, the second reason is even more important: if the mesh is very fine, then it becomes practically impossible to supply sufficiently accurate meteorological data for the model (to achieve this one must have too many meteorological stations and/or very high resolution meteorological forecast models).

Some kind of optimality has to be achieved: the numerical algorithms that are implemented in an LRTAP model should be sufficiently accurate (in relation to the uncertainties in the input data and the physical parameters), but the accuracy must not be much higher than what is really needed (very high accuracy of the numerical methods will certainly cause a lot of difficulties when the LRTAP model is treated numerically, and it will also be difficult to supply sufficiently accurate input data in this situation). The ideal solution will be to implement automatic error estimators in the models by which the errors due to the numerical methods used could be kept in a robust and reliable way under the desired level. However, the truth is that it is both difficult and expensive (in terms of computing time) to develop such automatic error estimators. In some models there are error estimators for some of the numerical methods used in the code (very often this is done for the time-integration parts; see, for example, Zlatev [515,517,518,520] or Zlatev et al. [532]). However, there is no model, until now at least, in which one tries to evaluate automatically the global error caused by the application of **all** numerical methods that are implemented in the model and to keep it in an adaptive way under some prescribed level.

The above discussion explains why sets of different numerical tests are necessary in order (i) to investigate what is the error due to the numerical algorithms used in the model and (ii) to give an answer to the question: is the model able (potentially, at least) to produce sufficiently accurate results? Several such tests will be described in this chapter.

4.1. Testing the advection algorithms

It has been stated several times in the previous chapters that the advection process (the transport due to the wind) is one of the most important processes and, therefore, it must be studied carefully. It is nearly obvious why the advection process is very important. Having the big emission sources given, one should like to know where the emitted pollutants are transported. In the same way, having some high concentrations in some part of the region, one should like to investigate how these have been transported there by the wind. Therefore the first tests concerning the reliability of the air pollution models are normally connected with checking the accuracy of the advection algorithms.

Assume that the model is a two-dimensional model. Then there is a rather simple and very suitable test-problem for the advection algorithms. The idea is very simple (see also Chapter 2). One defines a velocity field in which the trajectories of the winds are concentric circles around the centre of the space domain. Moreover, the velocity field is defined so that the angular velocity is constant. Assume now that the concentrations in the whole space domain G , excepting some sub-domain $G_1 \subset G$, are equal to some background concentration c_0 . Assume also that the concentrations in G_1 are different from c_0 (say, much higher than c_0). If the above conditions are satisfied, then it is clear that after some prescribed time t_1 (which can be calculated exactly) the parcels containing high concentrations will accomplish a full rotation around the centre of the space domain and will return at the same place where these were at the starting time t_0 . The advantages of using such a test-problem are clear:

(1) its analytic solution is known,

(2) the trajectories, along which the concentrations are transported, are not trivial (the transport is not carried out along straight lines, but along circles).

Therefore such a test-problem is very useful and the accuracy of the advection algorithms that are implemented in the model could (and should) be checked by using such a test-problem.

The ideas sketched above can be described mathematically in the following way. Consider:

$$(4.1) \quad \partial c / \partial t = - (y-1) \partial c / \partial x - (1-x) \partial c / \partial y,$$

where

$$(4.2) \quad x \in [0, 2], \quad y \in [0, 2], \quad t \in [0, 2\pi].$$

Note that the wind velocities in (4.1), $u = y-1$ and $v = 1-x$, are very special. When the wind velocities are defined in this way, the requirements stated above are satisfied. That is, (1) the wind trajectories are circles around the centre of the domain ($x_1=1, y_1=1$) and (2) the motion is performed with a constant angular velocity.

The initial distribution, $c(x,y,0)$, can be defined in many different ways. The distribution given below has been used in many papers; see, for example, Crowley [81], Molenkamp [320], Zlatev et al. [529], as well as the references in the last paper.

Consider the space domain that is defined by (4.2) and assume that

$$(4.3) \quad x_0 = 0.5, \quad y_0 = 1.0, \quad r = 0.25,$$

$$r^* = [(x-x_0)^2 + (y-y_0)^2]^{0.5}.$$

Then the initial distribution of the concentrations can be introduced by

$$(4.4) \quad c(x,y,0) = c_0 + 99c_0(1-r^*/r) \quad \text{as} \quad r^* < r \quad \text{and} \quad c_0 > 0$$

on G_1 and by

$$(4.5) \quad c(x,y,0) = c_0 \quad \text{as} \quad r^* \geq r,$$

on $G - G_1$, where c_0 is some background concentration; $c_0 = 0$ is often used in the experiments. If $c_0=0$, then $100(1-r^*/r)$ should be used in the right-hand side of (4.4).

Let us assume that the space domain G is discretized by using a (96x96) grid (such a discretization is used in the new version of **the Danish Eulerian Model**, and this is the reason for using it also in this chapter). If the surface defined by the initial distribution is plotted, then the result is a cone in the sub-domain G_1 (see Fig. 4.1). The base of the cone is a circle with centre (x_0, y_0) and a radius r . The centre of the cone is rotated along a circle with centre (x_1, y_1) and radius 0.5. The shape of the cone is preserved during the rotation. Therefore the test is often called a **cone rotation**. One can also define the initial distribution in many other ways. The distribution which is known under the name "**the cosine hill rotation**" is often used in practice. The cosine hill is defined by

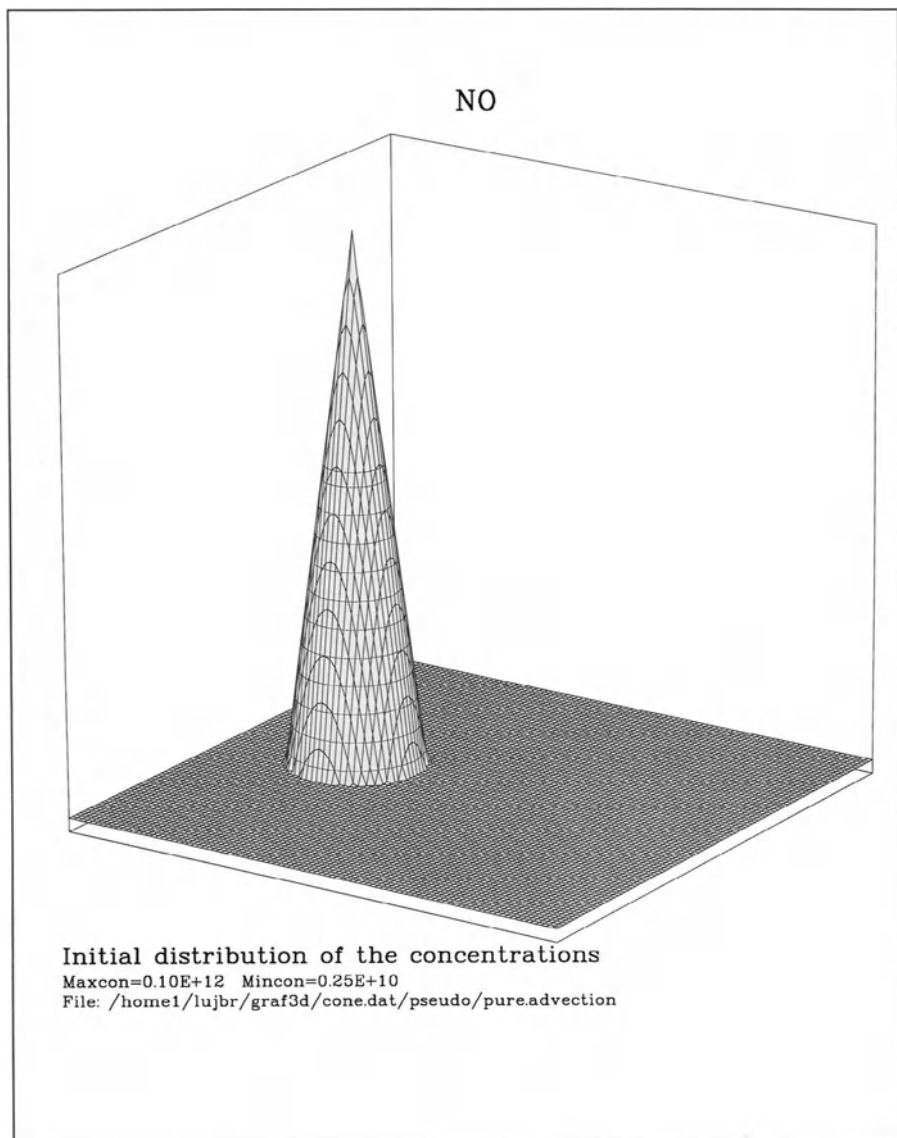


Figure 4.1
The initial distribution of the NO concentrations
(with background concentrations 2.5% of the maximum).

$$(4.6) \quad c(x, y, 0) = 2c_0 + 49[1 + \cos(\pi r^*/r)] \quad \text{as} \quad r^* < r \quad \text{and} \quad c_0 > 0$$

and by (4.5) as $r^* \geq r$ (see, for example, Zlatev et al. [529] and the references given there). If $c_0 = 0$, then $50(1 + \cos(\pi r^*/r))$ should be used in the right-hand side of (4.6).

The cosine hill rotation test is less stringent than the cone rotation. A test that is more stringent than the cone rotation is described and tested in Odman and Russel [331,332]. The test defined by (4.1) for any initial distribution of the concentrations is also called **a puff rotation**.

One can also introduce an emission source (or several emission sources) in the rotation test. This can be done by replacing (4.1) with

$$(4.7) \quad \partial c / \partial t = - (y-1) \partial c / \partial x - (1-x) \partial c / \partial y + E(x, y, t).$$

The initial distribution of the concentrations can be defined by

$$(4.8) \quad c(x, y, 0) = c_0 \quad \text{for} \quad \forall x \in [0, 2] \quad \text{and} \quad \forall y \in [0, 2]$$

when one or several sources are introduced in the space domain of the model.

Assume that only one source is to be used. Then it could be introduced as follows:

$$(4.9) \quad E(x, y, t) = 100a(1 - r^*/r) \quad \text{as} \quad r^* < r,$$

where a is a constant chosen so that the concentrations at the top of the plume have the same order of magnitude as the concentration at the top of the cone in the previous test example, and

$$(4.10) \quad E(x, y, t) = 0 \quad \text{as} \quad r^* \geq r.$$

Again the source (or the sources) can be defined in different ways. This test is called **a plume rotation**.

The pseudospectral method performs very well on both tests. Results that are obtained after one rotation of the cone are given in Fig. 4.2. Results obtained after one rotation of the plume defined by (4.7)-(4.10) are given in Fig. 4.3.

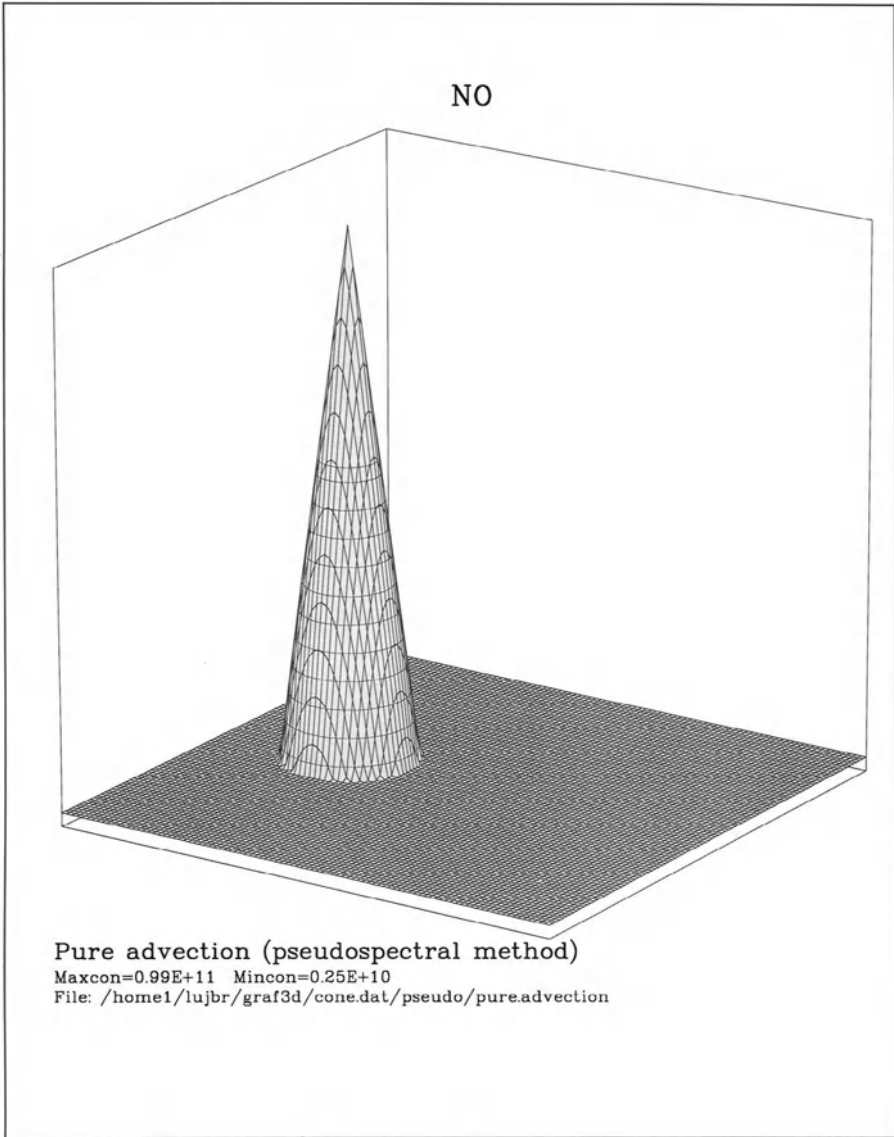


Figure 4.2
Advection of a puff treated by the pseudospectral method.
The NO concentrations after one rotation.

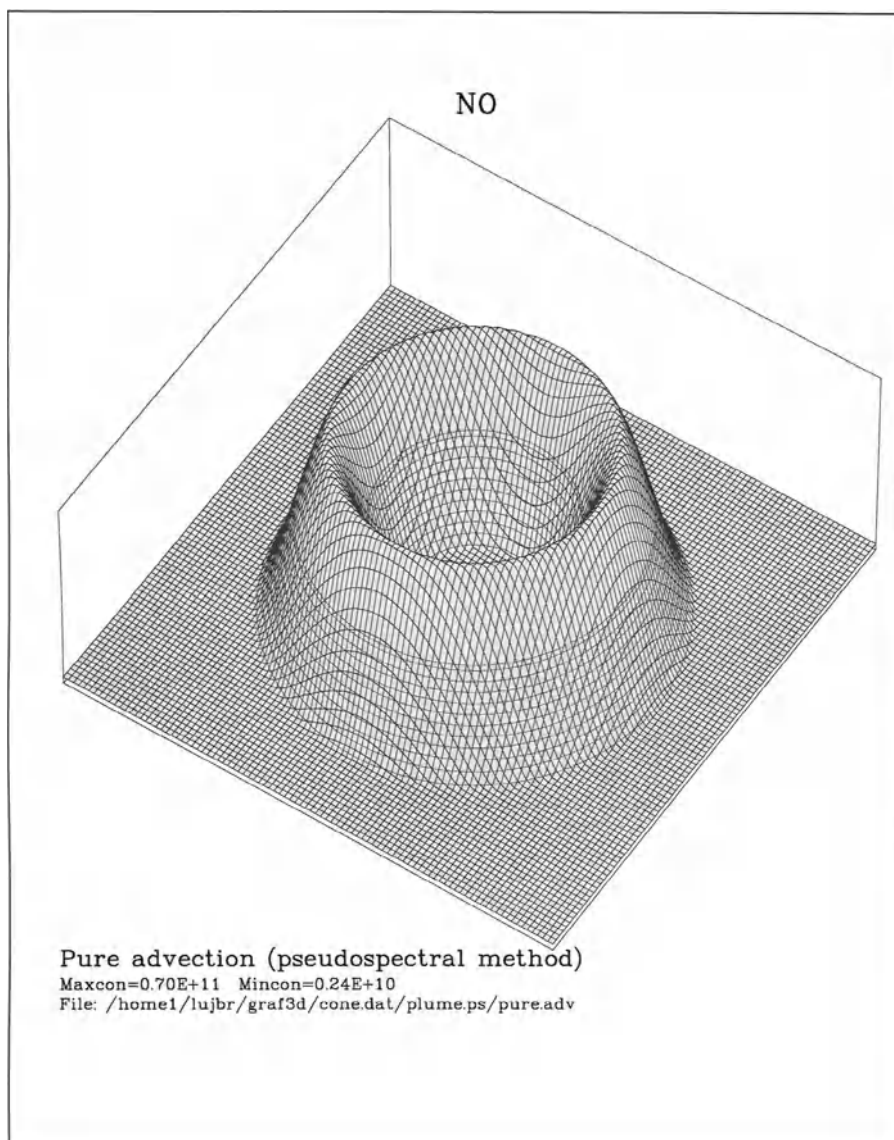


Figure 4.3
Advection of plume by the pseudospectral method.
The NO concentrations after one rotation.

The one-dimensional linear finite elements perform also quite good. Some results are given in Fig. 4.4 (the cone rotation) and Fig. 4.5 (the plume rotation).

It must be emphasized that several implementation problems have been omitted in order to facilitate the explanation of the main ideas. For example, periodic boundary conditions are necessary when the pseudospectral method is selected. For this particular problem this does not cause any problem (because the concentrations on the boundaries are periodical). If the linear finite elements are in use, then no boundary condition is used at out-flow points, while the values of the concentrations in in-flow points are set equal to the background concentration c_0 .

It should also be mentioned that if a pure advection test is carried out, then the negative concentrations (if such concentrations appear) are small and remain small during the whole integration. Therefore they are tolerated (no smoothing during the computational process is carried out). However, the final fields are smoothed by using a passive filter very similar to that proposed by Forester [168] (see also McRae et al. [315]). Smoothing becomes much more important when non-linear chemical reactions are coupled to the test; this case will be discussed in the next section.

4.2. Testing the numerical algorithms used in the combination of the advection and chemistry sub-models

Although the advection is a very important process, it is also important to study the transformations that take place under the transport. The chemical reactions are the most important source for such transformations (for some of the pollutants, at least). Therefore both the accuracy of the numerical algorithms that are used in the chemical sub-model and the accuracy of the coupling between advection and chemistry must carefully be tested.

The test problems that involve chemical reactions are much more complicated than the test problems that were discussed in the previous section. Only one PDE was sufficient to test the accuracy of the advection process. If the numerical algorithms that handle the chemical reactions are to be studied, then a system of PDE's must be used in the tests.

It is important to construct a test that contains as small number of PDE's as possible and, at the same time, all difficulties for the numerical procedures must be preserved in the chemical scheme used in the test. The last condition implies that the chemical scheme must contain non-linear chemical reactions (including here photochemical reactions). The simplified chemical scheme (which has been described in Section 2.6) satisfies this condition and, therefore, it can successfully be used in test problems.

A class of test problems that can be used to test both the accuracy of the algorithms used in the chemical sub-model and the coupling between advection and chemistry can be introduced by the following system of PDE's:

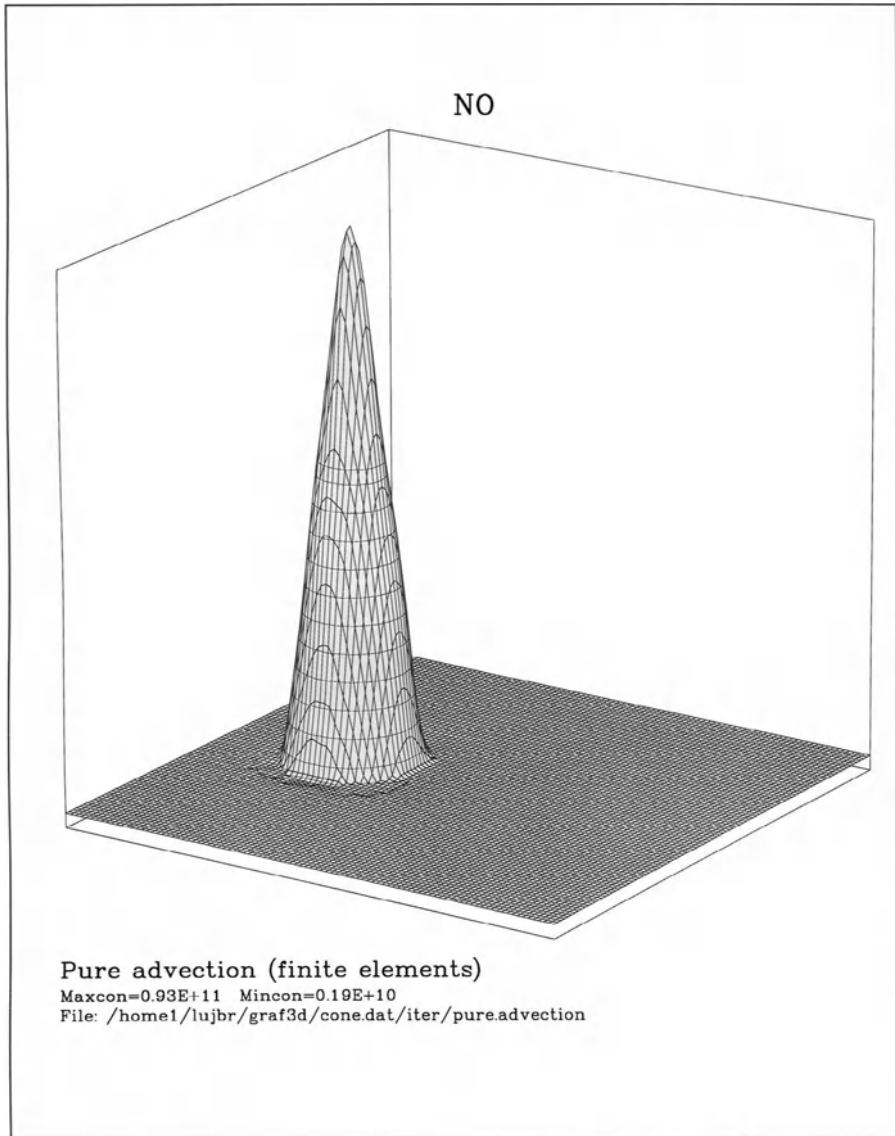


Figure 4.4
Advection of puff with the finite elements method.
The NO concentrations after one rotation.

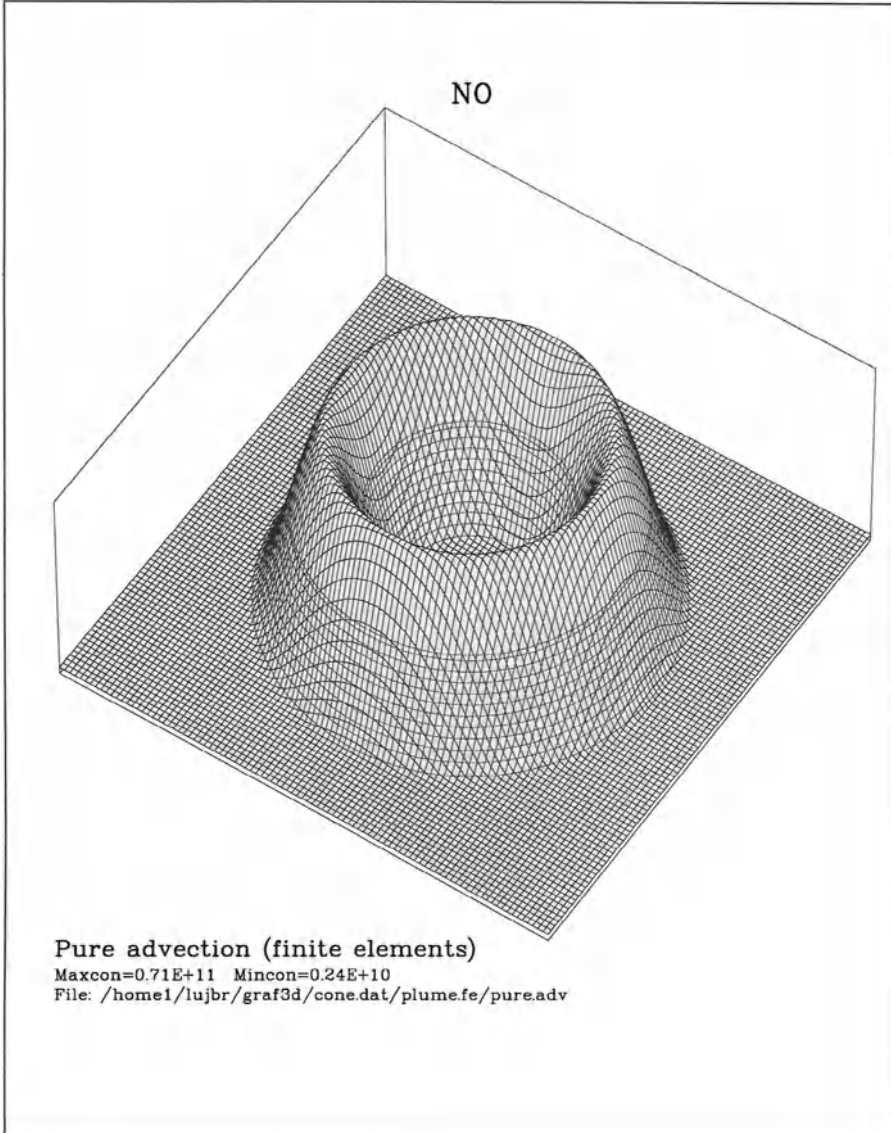


Figure 4.5.
Advection of plume by the finite elements method.
The NO concentrations after one rotation.

$$\begin{aligned}
 (4.11) \quad \partial c_s / \partial t = & - (y-1) \partial c_s / \partial x - (1-x) \partial c_s / \partial y \\
 & + Q_s(x, y, t) + E_s(c_1, c_2, \dots, c_q), \\
 & s = 1, 2, \dots, q, \quad q = 10,
 \end{aligned}$$

where the independent variables x , y and t are defined as in the previous section; see (4.2).

The class of test problems defined by (4.11) can be applied to study the following five types of tasks.

1. Pure advection of high concentrations (puffs). This type of tasks can be obtained from the system (4.11) by retaining only the advection terms, i.e. by setting both $E_s(x, y, t) = 0$ and $Q_s(c_1, c_2, \dots, c_q) = 0$. In this case it is sufficient to retain only one equation (see the previous section, where test problems for treating this type of tasks were studied).

2. Pure chemistry. Only the terms $Q_s(c_1, c_2, \dots, c_q)$ are retained in (4.11) when this type of tasks are to be studied. The transformations of the concentrations are performed **in situ** (the transformations of the concentrations in any point of the space domain are independent of the transformations in the other points).

3. Advection of high concentrations (puffs) + chemistry. Both the advection terms and the chemical terms are retained in (4.11) when this type of tasks are to be studied, but all emission sources are switched off by setting $E_s(x, y, t) = 0$. In this case the concentrations are transported along spherical trajectories with a constant angular velocity (see the previous section) and chemical reactions take place under the transport. Thus, this is an important type of tasks that allows us to test the accuracy of the splitting procedure.

4. Pure advection of plumes. Both the advection terms and the emission sources are retained in (4.11) when this type of tasks is to be studied, but the chemical reactions are switched off by setting $Q_s(c_1, c_2, \dots, c_q) = 0$. The pure advection of the concentrations caused by the emission sources (the plumes) can be followed when this is done. In this case it is sufficient to consider only one equation (see the previous section, where test problems for treating this type of tasks were studied).

5. Advection + sources + chemistry. All terms in (4.11) are retained when this type of tasks is to be studied. This means that this is the most general case that can be treated by (4.11).

The first and the fourth types of task have been discussed in the previous section. The other types will be discussed in this section.

Consider first the second type of tasks. Assume that the simplified chemical scheme from Chapter 2 is used. Let the initial concentrations be given by

$$(4.12) \quad c_s(x, y, 0) = c_{s0} + 99(1-r^*/r)c_{s0} \quad \text{as} \quad r^* < r,$$

$$(4.13) \quad c_s(x, y, 0) = c_{s0} \quad \text{as} \quad r^* \geq r$$

for the first four pollutants (NO, NO₂, HC and ALD), and by

$$(4.14) \quad c_s(x, y, 0) = c_{s0} \quad \text{for} \quad \forall x \in [0, 2] \quad \text{and} \quad \forall y \in [0, 2],$$

where $c_{s0} > 0$ is some background concentration of the s 'th pollutant, $s=1,2,\dots,10$, which depends on the particular pollutant. It is measured in mol/cm³, molecules per cubic centimetre, in the experiments that will be reported in this section.

Assume as in the previous section that the test problem (4.11) is discretized on a (96x96) grid and that the chemical sub-model is integrated over a time-interval that corresponds to the time-interval needed to perform a full rotation when only the advection sub-model is run; i.e. the time-interval is $[0, 2\pi]$. Then the chemical reactions depend only on the starting concentrations at the grid point under consideration. It is obvious that there are only a few grid-points with different sets of starting concentrations. This means that an accurate solution can be calculated (by using a very small time stepsize) at the few grid-points where the starting concentrations are different. After this a good approximation to the solution on the whole space domain can easily be constructed. Thus, although the exact solution is not known, a sufficiently accurate approximation can easily be obtained. By using this procedure one can easily verify that the solution shown in Fig. 4.6 is very accurate.

Consider the third type of tasks (the combined effect of advection and chemistry). Let us assume again that the initial concentrations are given by (4.12)-(4.14). Then the solution after one rotation should be the same as that shown in Fig. 4.6 (but note that it is not obtained in the same way: the transformations are performed in situ when only the chemical model is run, while the initial concentrations are transported along circular trajectories and the chemical transformations are performed under the transport when both advection and chemistry take place). Experiments show that again the pseudospectral method performs better than the finite elements; compare the results shown in Fig. 4.7 and Fig. 4.8. However, also in this case the results obtained with the finite elements algorithm are quite acceptable.

The numerical results that are shown in Fig. 4.7 and Fig. 4.8 are for nitrogen oxide, NO. This pollutant causes the greatest problems for the numerical methods, because its concentrations vary in the most widest range (among the pollutants involved in the simplified chemical scheme). The largest value of the NO concentrations is of order 10¹¹ mol/cm³, while its minimal value is practically equal to zero. The numerical results for the other pollutants are considerably more accurate.

The results shown in Fig. 4.7 and Fig. 4.8 are considerably better than the results published in Hov et al. [246]. The improvement was achieved by using both a finer space discretization and a filter, which smooths short waves. The filter is based on ideas described in Forester [168] and McRay et al. [315]. It is used passively. This means that only the output fields are smoothed by using the filter, i.e. results that will be used in further computations are never smoothed. This is an attempt to avoid accumulating effects of the filter on the final results. The filter works rather well for this kind of problems. However, it is not very clear if such a filter should be used in big air pollution models. In fact, such a filter is not used yet in **the Danish Eulerian Model** (Zlatev et al. [538,539]). Much more experiments are needed in order to find out if such a filter can be applied in more general situations and, if it can be used only in some special situations, how to determine the cases where the use of such a filter is worthwhile. Thus, the questions **when** and **how** to use a filter are open, and further research in this field, with a special emphasis on big air pollution models, is highly desirable.

One can also carry out **advection + chemistry** computations in the case where there is a source in the space domain (to rotate a plume by performing simultaneously the appropriate chemical reactions). Assume that the initial concentrations of every pollutant are given by (4.8). Assume that sources are introduced for NO, NO₂, CO and ALD by the relations (4.9)-(4.10). The test-problem defined as sketched above has been run both with the pseudospectral method and with the chapeau finite elements. The results obtained with the pseudospectral method for the nitrogen oxide concentrations after one rotation are given in Fig. 4.9. The corresponding results for the linear one-dimensional finite elements are given in Fig. 4.10.

A filter was also used in the plume rotation tests. The conclusions are the same as those for the puff rotation:

- (i) for these particular test problems the filter performs quite well,
- (ii) it is not quite clear when and how to use a filter

and

- (iii) further research in this field is necessary.

Again, as in the previous sections, some details have been omitted in an attempt to facilitate the explanation of the main ideas. For example, periodic boundary conditions are used when the pseudospectral method is used. For this particular problem this does not cause any problem (because the concentrations on the boundaries are periodical).

If the linear finite elements are in use, then no boundary condition is used at out-flow points, while the value of the concentrations at in-flow points is set equal to the background concentration $c_{s0}(t)$, where $c_{s0}(t)$, $s=1,2,...,q$, is calculated by running a box-model with the simplified chemical scheme using as starting values the background concentrations c_{s0} .

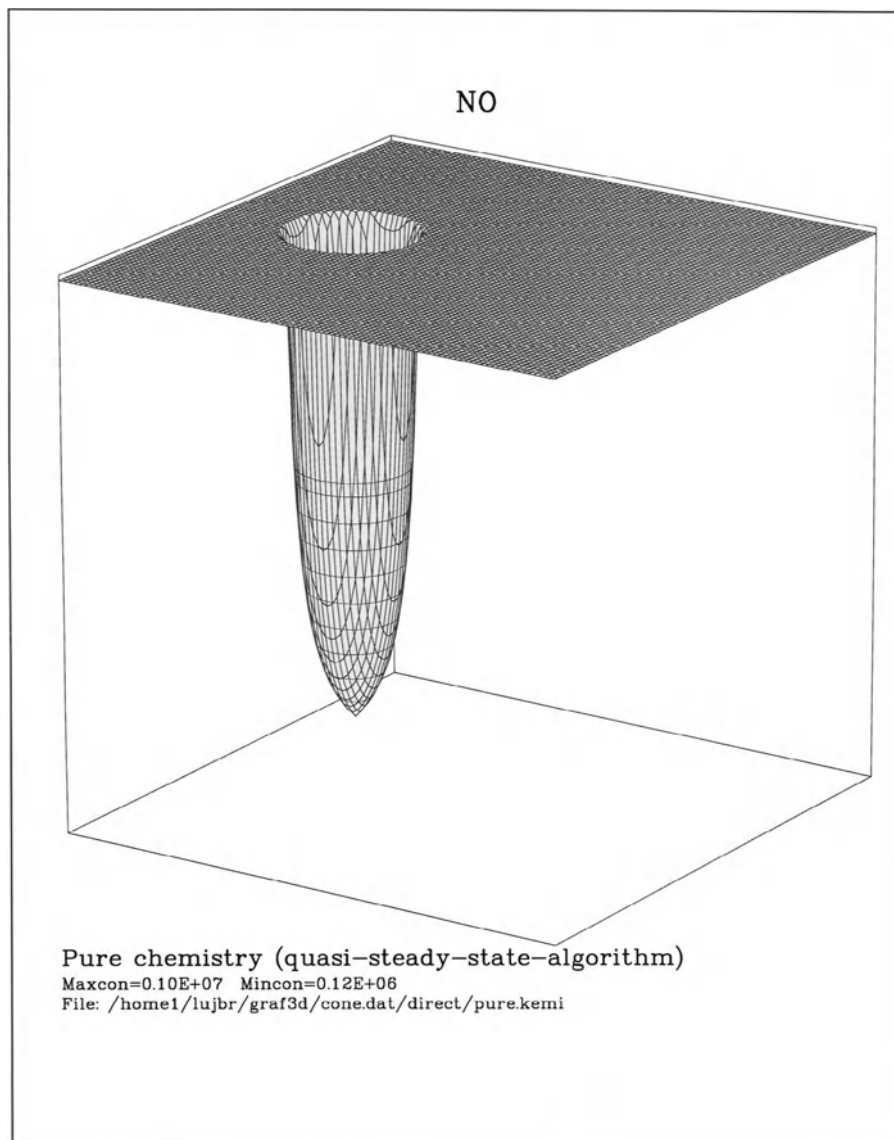


Figure 4.6
Chemistry performed on the concentrations shown in Fig. 4.1.
The time-interval is equal to that needed for one rotation.

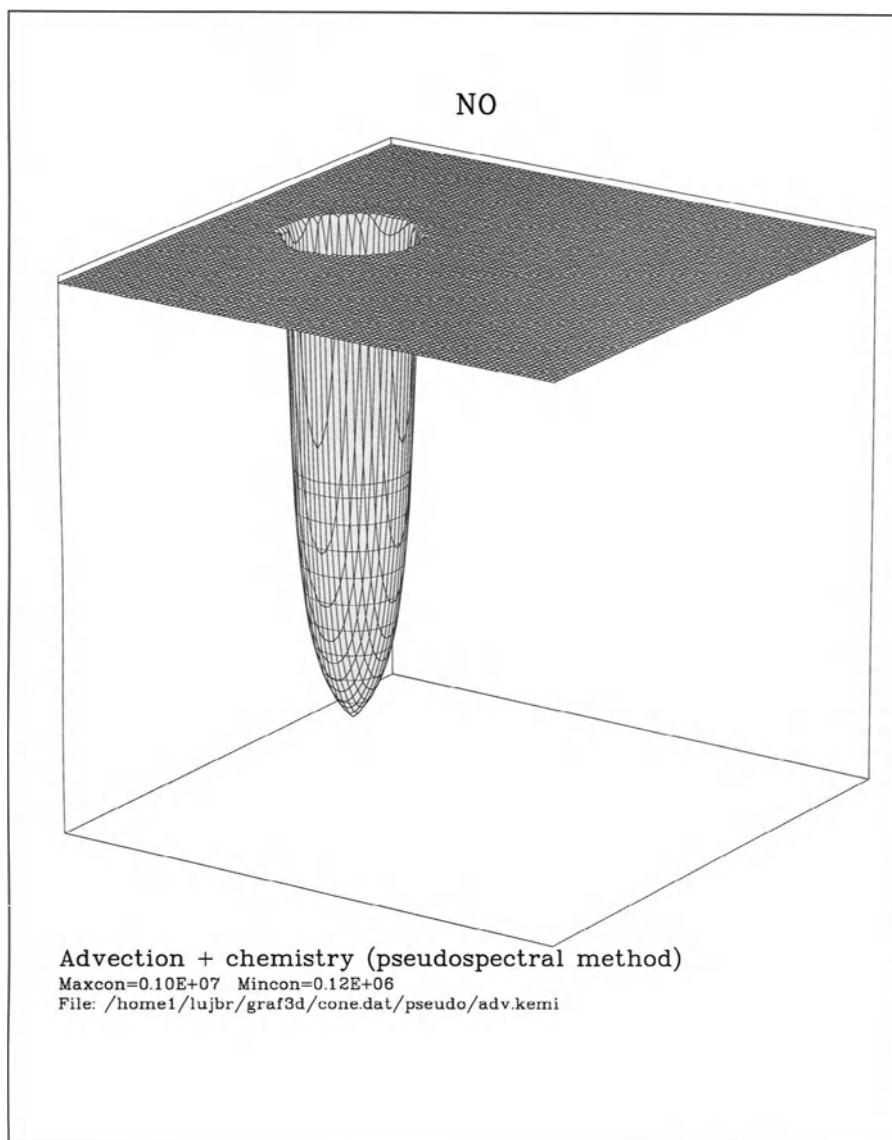


Figure 4.7
Advection + chemistry for a puff with the pseudospectral method. The
NO concentrations after one rotation.

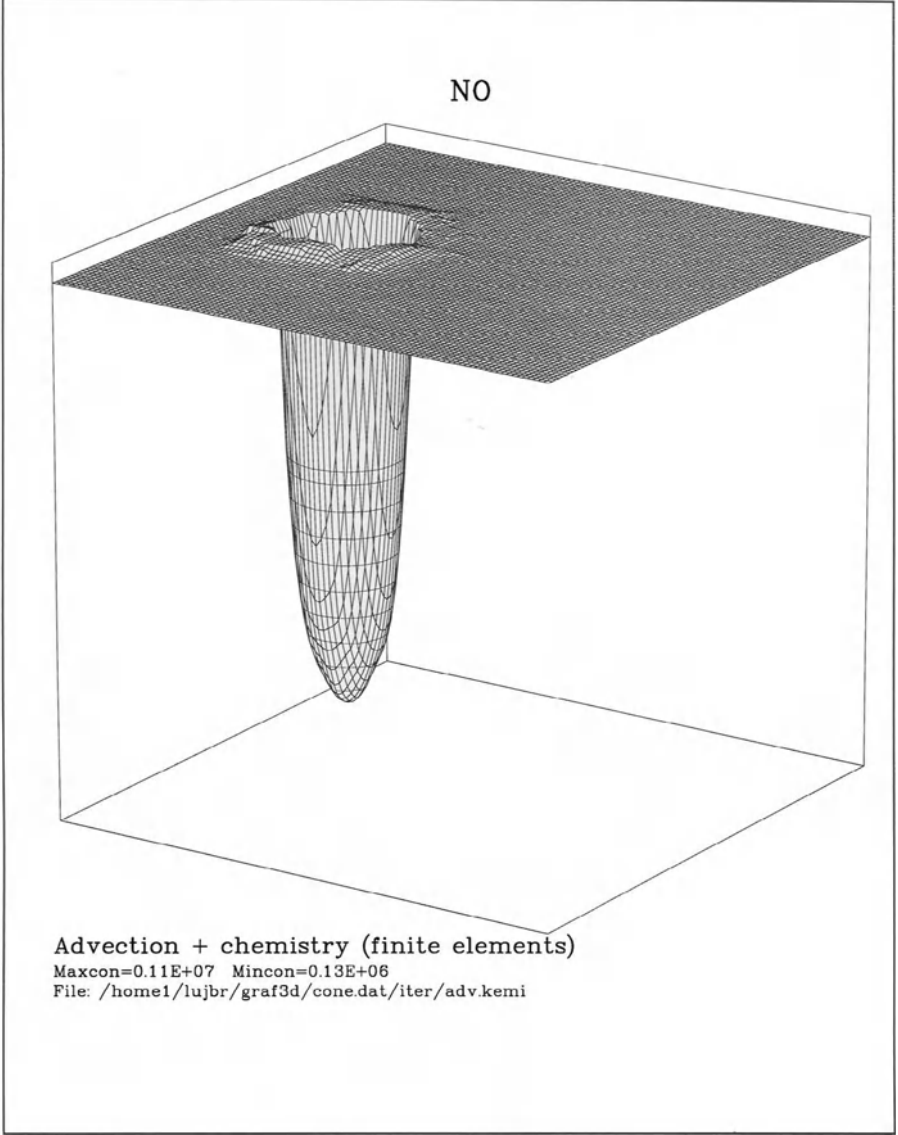


Figure 4.8.
Advection + chemistry for a plume with the finite elements method. The
NO concentrations after one rotation.

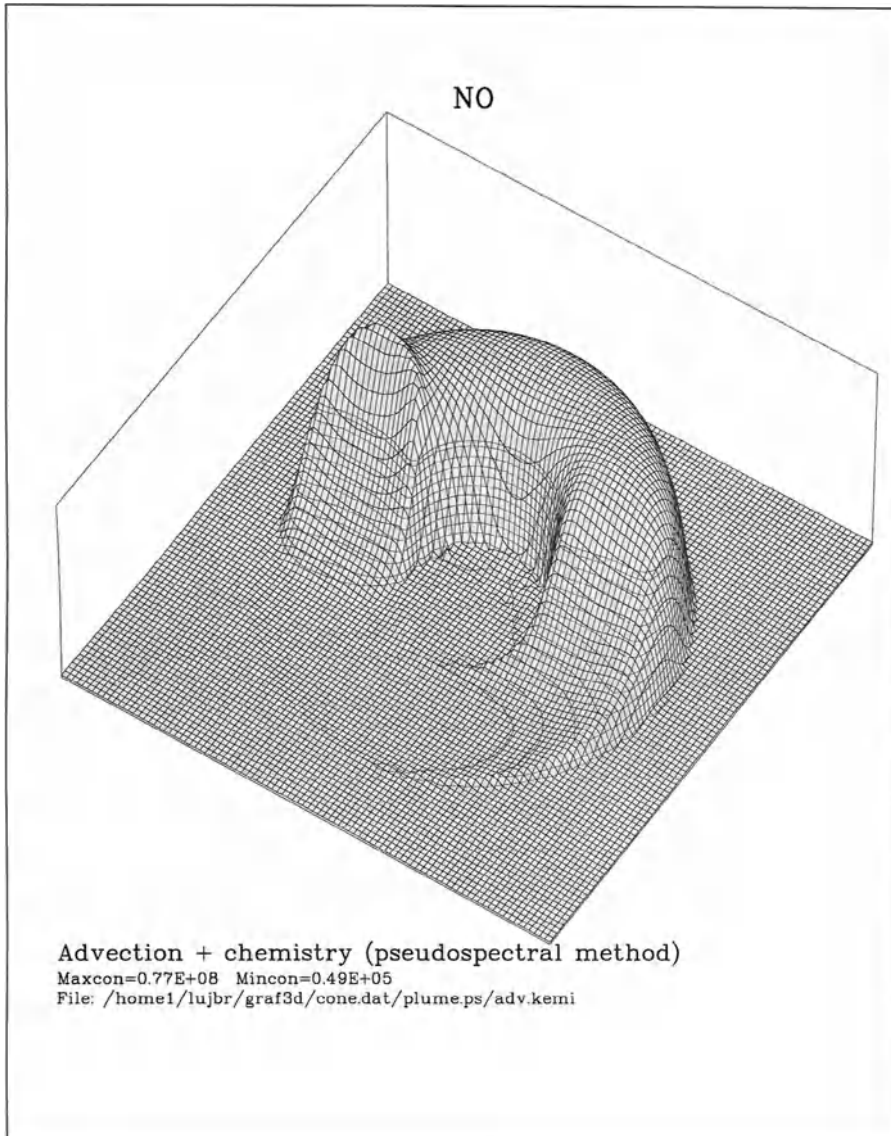


Figure 4.9
Advection + chemistry for a plume with the pseudospectral method. The
NO concentrations after one rotation.

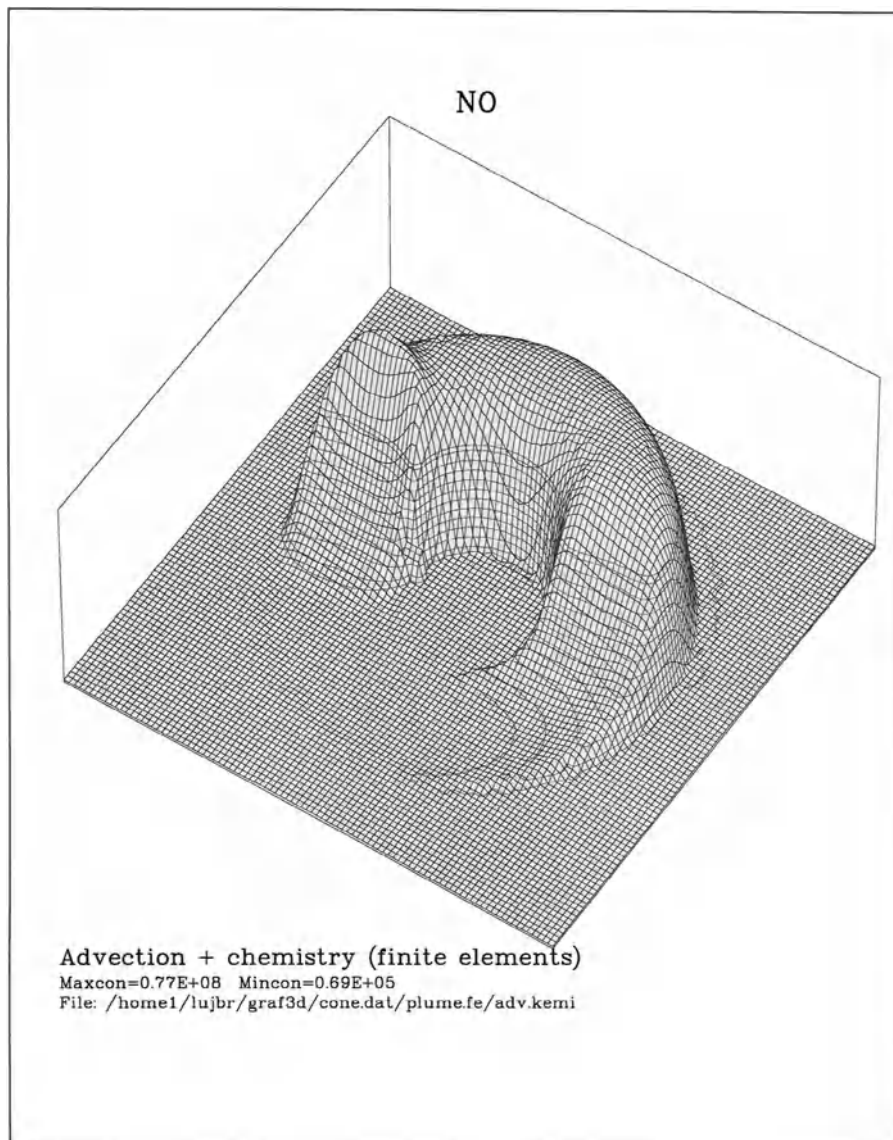


Figure 4.10.
Advection + chemistry for a plume with the finite elements method. The
NO concentrations after one rotation.

It should also be mentioned that if the chemical module is activated, then negative concentrations (if such concentrations appear) cannot be tolerated; these will cause severe instability problems. Therefore negative concentrations are smoothed by using the simple algorithm proposed in Hov et al. [246]. Only concentrations in an neighbourhood of a point where negative concentrations were detected are smoothed by this algorithm. The final fields are also smoothed by using a passive filter (this procedure has been discussed above), which is very similar to the filter that has been proposed by Forester [168] and discussed in McRae et al. [315]).

4.3. Other tests of numerical algorithms

The test problems that were described in the previous two sections are perhaps the most relevant tests for the most important components of a two-dimensional air pollution problem; the advection process and the chemical reactions. However, other test problems could also be very useful. There are at least three cases which require some additional test problems.

(i) Information that is worthwhile could sometimes be obtained from one-dimensional problems. However, one should be careful, because the transition from a one dimensional problem to a two-dimensional problem may lead to some difficulties, which are not present when the one dimensional case is studied.

(ii) The diffusion phenomena also may deserve some more attention. This is especially true when the grid-size is refined, because the artificial diffusion (that is caused by the fact that the emission are equally distributed on the whole grid-unit) is decreased considerably in this situation and, thus, one should take care to describe more accurately the diffusion process in a model that is to be treated on a fine grid.

(iii) Finally, three-dimensional models are often used now when fast modern computers are easily available. The transition from a two-dimensional model to a three-dimensional model may cause some difficulties that do not appear in the two-dimensional models. Therefore three-dimensional test problems are highly desirable.

Several test problems, which can be used to check the accuracy of the approximations in the three cases, which were sketched above, will be shortly discussed in this section.

4.3.1. Pure advection in one dimension

This process can be studied by considering the simple equation:

$$(4.15) \quad \partial c / \partial t = - \partial c / \partial x, \quad x \in [0, 2], \quad t \in [0, 1],$$

with a constant wind velocity $u=1$ and with the following initial condition

$$(4.16) \quad c(x, 0) = \sigma^{-1} \exp[-(x-x_0)^2/2\sigma^2],$$

where $x_0 = 0.5$ and different values of σ in the interval $[0.001, 0.1]$ can be used. The exact solution of this test problem is

$$(4.17) \quad c(x, t) = \sigma^{-1} \exp[-(x-x_0-t)^2/2\sigma^2].$$

Numerical results obtained in the solution of this test problem by the pseudospectral method are given by Zlatev et al. [530].

4.3.2. Advection + diffusion in one dimension

This process can be studied by using the following one dimensional test-equation

$$(4.18) \quad \partial c / \partial t = - u \partial c / \partial x + K_x \partial^2 c / \partial x^2,$$

$$x \in [0, 2], \quad t \in [0, 1],$$

with $u = 1$, $K_x = 0.1$ and with an initial value condition defined by

$$(4.19) \quad c(x, 0) = p_0$$

for $x = x_0$ and $c(x, 0) = 0$ otherwise. It is again assumed that $x_0 = 0.5$ and $p_0 = 100$.

It can be shown, see Berkowicz and Prahm [27], that the solution of the problem in the point $(1, 5, 1.0)$ is given by

$$(4.20) \quad c(x_0+1, 1) = 0.5 p_0 / (\pi K_x)^{0.5} \approx 89.206.$$

Numerical results obtained in the solution of this test problem by the pseudospectral method are given by Zlatev et al. [530].

4.3.3. Two dimensional transport in the Oxz plane with a vertical diffusion and a variable mixing height

This is a physical process which is close to some particular conditions under which the transport of air pollutants may take place. However, it should be emphasized here that while such a test-problem can successfully be used to study some physical processes under some special conditions (the horizontal component of the wind does not change with the height), it is completely wrong to use such kind of equations in a general-purpose model. The process that will be studied in this sub-section is described by the equation:

$$(4.21) \quad \partial C / \partial t = -u \partial C / \partial x + K_x \partial^2 C / \partial x^2 + \partial / \partial z (K_z \partial C / \partial z) + E,$$

where

$$(4.22) \quad x \in [0, 16], \quad z \in [0, 16], \quad t \in [0, 16],$$

and

$$(4.23) \quad u = 1.0, \quad K_x = 0.02.$$

Let us assume that we want to study two situations: (i) a change from day conditions to night conditions and (ii) a change from night conditions to day conditions. In the first situation the day conditions are defined for $t < 5$ and the night conditions for $t \geq 5$. In the second situation the night conditions are defined for $t < 5$ and the day conditions for $t \geq 5$. Let us assume also that the mixing height during the day is given by $H(x, t) = h_{\text{day}} = 12$, while the corresponding value for the night conditions is $H(x, t) = h_{\text{night}} = 4$. Then the vertical diffusion coefficient $K_z = K_z(x, z, t)$ depends on the mixing height $H(x, t)$ and can be defined by

$$(4.24) \quad K_z = 40 \quad \text{for} \quad z \leq H(x, t) \quad \text{and} \quad K_z = 0 \quad \text{otherwise.}$$

Assume now that the problem is discretized by

$$(4.25) \quad \Delta x = 0.5, \quad \Delta z = 1.0, \quad \Delta t = 2^{-4}.$$

The numbers of grid-points along the axes Ox and Oz and along the time direction are

$$(4.26) \quad NX = 32, \quad NZ = 17 \quad \text{and} \quad NT = 160.$$

Two kinds of sources were used: (i) low sources and (ii) high sources. A low source is defined by

$$(4.27) \quad E = 400/(\Delta x \Delta z) \quad \text{for} \quad x=3 \wedge z=2 \quad \text{and} \quad E = 0 \quad \text{otherwise.}$$

A high source is defined by

$$(4.28) \quad E = 400/(\Delta x \Delta z) \quad \text{for} \quad x=3 \wedge z=6,$$

$$E = 100/(\Delta x \Delta z) \quad \text{for the eight grid points that are}$$

$$\text{adjacent to } (x, z) = (3, 6),$$

$$E = 0 \quad \text{otherwise.}$$

The initial condition is given by

$$(4.29) \quad c(x, z, 0) = 0.$$

The boundary conditions are defined as Dirichlet boundary conditions (the concentrations on the boundaries are given) by

$$(4.30) \quad c(0, z, t) = c(16, z, t) = 0$$

$$\text{for} \quad \forall z \in [0, 16] \quad \wedge \quad \forall t \in [0, 10]$$

and

$$(4.31) \quad c(x, 0, t) = c(x, 16, t) = 0$$

$$\text{for} \quad \forall x \in [0, 16] \quad \wedge \quad \forall t \in [0, 10].$$

This test problem is described in detail in Zlatev et al. [531]. Numerical results are also given there. Several different situations can be studied qualitatively by the problem defined by (4.21)-(4.31). It is quite sufficient to emphasize once again the following three important properties of this test problem.

The first property of the test-problem defined by (4.21)-(4.31) is connected to the mixing height: the vertical diffusion coefficient $K_z = K_z(x, z, t)$ depends on the mixing height. K_z is positive when z is less than the mixing height and zero otherwise. Moreover, in the test it is assumed that the mixing height during the day is higher than that during the night.

The second property of the test-problem defined by (4.21)-(4.31) is connected with the study of the interactions between sources and mixing height. Four typical situations were studied in Zlatev et al. [531]:

- (a) a low source together with changing from night conditions to day conditions,
- (b) a low source together with changing from day conditions to night conditions,
- (c) a high source together with changing from night conditions to day conditions,
- (d) a high source together with changing from day conditions to night conditions.

The third property of the test-problem defined by (4.21)-(4.31) concerns the information about the solution. The exact solution is not known in this situation. However, some useful information about the qualitative behaviour of the solution can be obtained by using physical reasons; see again Zlatev et al. [531].

4.3.4. Three dimensional transport with variation of the mixing height and the wind velocities

The governing equation for this process is given by

$$(4.32) \quad \frac{\partial c}{\partial t} = -u \frac{\partial c}{\partial x} - v \frac{\partial c}{\partial y} + K_x \frac{\partial^2 c}{\partial x^2} + K_y \frac{\partial^2 c}{\partial y^2} + \frac{\partial}{\partial z} (K_z \frac{\partial c}{\partial z}) + E,$$

where

$$x \in [0, 16], \quad y \in [0, 16], \quad z \in [0, 16], \quad t \in [0, 16].$$

It is assumed that the horizontal diffusion coefficients are positive constants that are given by

$$(4.33) \quad K_x = 0.02, \quad K_y = 0.02.$$

Let us assume that we want to study the situation where **a change from day conditions to night conditions takes place**. Moreover, let the day conditions be defined for $t < 5$ and night conditions for $t \geq 5$. Then the wind velocities are defined as follows:

$$(4.34) \quad u = 1.0 \quad \text{for} \quad t < 5 \quad \text{and} \quad u = 2^{-0.5} \quad \text{otherwise},$$

$$(4.35) \quad v = 0.0 \quad \text{for} \quad t < 5 \quad \text{and} \quad v = 2^{-0.5} \quad \text{otherwise}.$$

Assume also that the mixing height is given by $H(x, y, t) = h_{\text{day}} = 12$ during the day, while the corresponding value of the mixing height for the night conditions is given by $H(x, y, t) = h_{\text{night}} = 4$. Then the vertical diffusion coefficient $K_z = K_z(x, y, z, t)$ will depend on the mixing height $H(x, y, t)$. Moreover, it will be assumed here that the vertical diffusion is greater during the day. The precise definition of K_z is given below

$$(4.36) \quad K_z = 40 \quad \text{for} \quad z \leq H(x, y, t) \quad \wedge \quad t \leq 5,$$

$$K_z = 10 \quad \text{for} \quad z \leq H(x, y, t) \quad \wedge \quad t > 5,$$

$$K_z = 0 \quad \text{otherwise}.$$

Assume now that the problem is discretized by

$$(4.37) \quad \Delta x = 0.5, \quad \Delta y = 0.5, \quad \Delta z = 1.0, \quad \Delta t = 2^{-4}.$$

The numbers of grid-points along the axes Ox and Oz and along the time direction are

$$(4.38) \quad NX = 32, \quad NY = 32, \quad NZ = 17 \quad \text{and} \quad NT = 160.$$

Two kinds of emission sources were used in the experiments: (i) low sources and (ii) high sources. A low source is defined by

$$\begin{aligned}
 (4.39) \quad E &= 4000 / (\Delta x \Delta y \Delta z) \quad \text{for} \quad (x, y, z) = (3, 3, 2), \\
 &\quad (x, y, z) = (2.5, 3, 2), \\
 &\quad (x, y, z) = (3, 2.5, 2), \\
 &\quad (x, y, z) = (2.5, 2.5, 2), \\
 \\
 E &= 2000 / (\Delta x \Delta y \Delta z) \quad \text{for} \quad (x, y, z) = (2.5, 2, 2), \\
 &\quad (x, y, z) = (3, 2, 2), \\
 &\quad (x, y, z) = (2.5, 3.5, 2), \\
 &\quad (x, y, z) = (3, 3.5, 2), \\
 &\quad (x, y, z) = (2, 2.5, 2), \\
 &\quad (x, y, z) = (2, 3, 2), \\
 &\quad (x, y, z) = (3.5, 2.5, 2), \\
 &\quad (x, y, z) = (3.5, 3, 2), \\
 \\
 E &= 1000 / (\Delta x \Delta y \Delta z) \quad \text{for} \quad (x, y, z) = (2, 2, 2), \\
 &\quad (x, y, z) = (2, 3.5, 2), \\
 &\quad (x, y, z) = (3.5, 2, 2), \\
 &\quad (x, y, z) = (3.5, 3.5, 2), \\
 \\
 E &= 0 \quad \text{otherwise.}
 \end{aligned}$$

A high source is defined by

$$(4.40) \quad E = 8000 / (\Delta x \Delta y \Delta z) \quad \text{for} \quad (x, y, z) = (3, 9, 8),$$

$$\begin{aligned}
 E = 4000 / (\Delta x \Delta y \Delta z) \quad \text{for} \quad & (x, y, z) = (3.5, 9, 8), \\
 & (x, y, z) = (2.5, 9, 8), \\
 & (x, y, z) = (3, 8.5, 8), \\
 & (x, y, z) = (3, 9.5, 8),
 \end{aligned}$$

$$\begin{aligned}
 E = 2000 / (\Delta x \Delta y \Delta z) \quad \text{for} \quad & (x, y, z) = (2.5, 8.5, 8), \\
 & (x, y, z) = (3.5, 8.5, 8), \\
 & (x, y, z) = (2.5, 9.5, 8), \\
 & (x, y, z) = (3.5, 9.5, 8),
 \end{aligned}$$

$$E = 0 \quad \text{otherwise.}$$

The initial condition is given by

$$(4.41) \quad c(x, y, z, 0) = 0.$$

The boundary conditions are again (as in the previous test problem) defined as Dirichlet boundary conditions:

$$\begin{aligned}
 (4.42) \quad & c(0, y, z, t) = c(16, y, z, t) = 0 \\
 & \text{for} \quad \forall y \in [0, 16] \quad \forall z \in [0, 16] \quad \wedge \quad \forall t \in [0, 10],
 \end{aligned}$$

$$\begin{aligned}
 (4.43) \quad & c(x, 0, z, t) = c(x, 16, z, t) = 0 \\
 & \text{for} \quad \forall x \in [0, 16] \quad \forall z \in [0, 16] \quad \wedge \quad \forall t \in [0, 10],
 \end{aligned}$$

$$\begin{aligned}
 (4.44) \quad & c(x, y, 0, t) = c(x, y, 16, t) = 0 \\
 & \text{for} \quad \forall y \in [0, 16] \quad \forall x \in [0, 16] \quad \wedge \quad \forall t \in [0, 10],
 \end{aligned}$$

This test problem is described in detail in Zlatev et al. [533]. Numerical results are given there. It should only be emphasized that with this test problem several situations (which are rather simple but, at the same time, also quite realistic) can be studied qualitatively.

4.3.5. Pure advection in three dimensions

The two test problems that were considered in the previous two sub-sections provide only a qualitative information about the solution. Such test problems are very useful, but test problems where the exact solution is known (or, at least, some sufficiently accurate approximation of the exact solution is available) are also highly desirable. A three dimensional pure advection test problem, whose exact solution is known, has been proposed by Christensen [70]. This problem is defined by

$$(4.45) \quad \partial c / \partial t = -(y_1 - y) \omega \partial c / \partial x - (x - x_1) \omega \partial c / \partial y - (x - x_1) \omega \partial c / \partial z,$$

where x_1 and y_1 are given, and the space domain is defined by

$$(4.46) \quad x \in [a_1, b_1], \quad y \in [a_2, b_2], \quad z \in [a_3, b_3], \\ t \in [a_4, b_4].$$

The exact solution of (4.45) is

$$(4.47) \quad c(x, y, z, t) = c((x - x_1) \cos(\omega t) + (y_1 - y) \sin(\omega t) + x_1, \\ - (y_1 - y) \cos(\omega t) + (x - x_1) \sin(\omega t) + y_1, \\ - (y_1 - y) \cos(\omega t) + (x - x_1) \sin(\omega t) + y_1 - (y - z), a_4).$$

It is convenient to take $(x_1, y_1, z_1) = (1, 1, 1)$, $\omega = 1$ and to define the space domain and the time-interval by

$$(4.48) \quad x \in [0, 2], \quad y \in [0, 2], \quad z \in [0, 2], \quad t \in [0, 2\pi].$$

It is clear that this is not a restriction (there exists a simple transformation of the independent variables by which this could be achieved). It is also clear that if the space domain is defined by (4.48), then (x_1, y_1, z_1) is the centre of the domain.

Assume that the initial condition is given by the following field of concentrations:

$$(4.49) \quad c(x, y, z, 0) = c_0 + 99c_0(1 - r^*/r) \quad \text{as} \quad r^* < r \quad \text{and} \quad c_0 > 0$$

and

$$(4.51) \quad c(x, y, z, 0) = c_0 \quad \text{as} \quad r^* \geq r,$$

where

$$(4.52) \quad x_0 = 0.5, \quad y_0 = 1.0, \quad z_0 = 1.0, \quad r = 0.25$$

and

$$(4.53) \quad r^* = [(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2]^{0.5}.$$

This test problem can be considered as a three-dimensional generalization of the two-dimensional pure advection test problem (4.1) that has been described in Section 4.1. However, the test-problem defined by (4.45)-(4.53) is in a way a much more stringent problem than (4.1), because the shapes of the iso-surfaces that are formed by the high concentrations change during the transport. The initial iso-surfaces formed by the high concentrations are spheres. Under the transport they are transformed into ellipsoids. After one full rotation the ellipsoids are transformed back to spheres. It is difficult for the numerical methods to re-shape the ellipsoids into spheres. Therefore this is a good test of the three-dimensional advection subroutines: the numerical algorithms that are applied in these subroutines must be able not only to follow the transport along the three-dimensional trajectories (which are in general ellipses), but also to transform correctly the shapes of the iso-surfaces.

4.3.6. Common remarks

The problems discussed in this section are test problems which deal only with the advection part and the advection-diffusion part of the models. However, it is not very difficult to extend these test problems so that also some of the remaining physical processes are involved. It is worthwhile to attach a chemical sub-models to these tests and/or to introduce sources in some parts of the space domain. Such extensions can be constructed by following closely the devices that were used in the test-problems discussed in the first two sections of this chapter. The last test-problem will give much more information about the reliability of the numerical method if both chemical reactions and sources are included in it.

The three dimensional test-problems are much more difficult than the two-dimensional test-problems. They produce huge sets of output data, and it is rather difficult to understand the tendencies and to judge correctly the ability of the numerical methods to produce sufficiently accurate results. A good graphic subroutines are absolutely necessary in order to represent the results in a form that allows us to see the performance of the numerical methods (and judge the accuracy of the results). However, even excellent visualization may not be sufficient in some situations, because it is also important to see the performance in some intermediate stages and to judge the accuracy of some intermediate results. Therefore animation of the computational process is the perfect means which allows us to understand best the performance of the numerical methods when three dimensional test-problems are used. It should be mentioned here that several animation programs have been developed at the National Environmental Research Institute and extensively used in the experiments with the test-problems (and also with the air pollution models; see also Chapter 10).

4.4. Concluding remarks and references concerning the tests of the reliability of the numerical algorithms

The requirement that the numerical methods (which are applied in the treatment of a big air pollution model) are sufficiently accurate is a basic requirement.

The errors due to the application of approximate algorithms must be less than the errors due to the uncertainties in the input data (both meteorological data and emission data) and in the determination of the parameters that are used to describe different physical processes involved in the models.

If this condition is satisfied, then one can, as mentioned several times in the previous chapters, study many important topics, such as

- (i) the sensitivity of the output results to changes of different components in the meteorological data,
- (ii) the influence of changes in the emission fields on the output results,
- (iii) the sensitivity of the results to changes of different mechanism by which the physical processes are described.

This explains why test problems have been extensively used during the last two decades in experiments with models. The pure advection test described in **Section 4.1** has been

proposed simultaneously in 1968 by Crowley [81] and Molenkamp [320]. After that this model and/or different modifications of this model have been used in many papers and books.

In the new LRTAP models advanced chemical reactions are as a rule incorporated. Therefore there is a need to check both the accuracy of the numerical algorithms by which the chemical reactions are treated and the coupling of the chemical sub-model with the other sub-models (the coupling with the advection sub-model being the most important one). Special tests in which the chemistry is also checked are discussed in **Section 4.2**. The combination of the rotation tests with tests of the chemical algorithms (and, what is even more important, with tests of the accuracy of the procedures used in the splitting of advection and chemistry) was probably first suggested in Hov et al. [246]. Now such a test has also been used by other authors; see, for example, Odman and Russell [331,332].

All tests in Section 4.1 and in Section 4.2 have been performed by using a (96x96) grid. However, it is worthwhile to mention here that the pseudospectral algorithm gives good results also when a (32x32) grid is used; and even if a (16x16) grid is applied; see, for example, Zlatev et al. [529]. Also the linear one-dimensional finite element method has been run on a (32x32) grid; see Zlatev and Wasniewski [549].

The tests described in Section 4.1 and Section 4.2 are very important, but some other tests are also needed (in some situations, at least). Other tests of the numerical algorithms that can be used in the models (including here tests concerning three-dimensional models) are shortly discussed in **Section 4.3**; these tests are fully described in Christensen [70] and Zlatev et al. [529-533].

CHAPTER 5

NEED FOR EFFICIENT ALGORITHMS

"To understand variable stepsize, variable formula (VSVF) codes, a crucial first step is to establish stability of the integration".

L. F. Shampine and W. Zhang: *"Rate of Convergence of Multistep Codes Started by Variation of Order and Stepsize"*, SIAM J. Numer. Anal., **27**(1990), 1506-1518.

The discretization of the systems of PDE's, by which the large LRTAP models are described mathematically, always leads to very big computational problems. Therefore it is crucial to perform efficiently the most time-consuming calculations during the numerical treatment of a large air pollution model. Efficiency can be achieved by selecting numerical algorithms which are both fast and accurate and by implementing correctly these algorithms in the codes by which the large air pollution models are treated numerically.

This way of achieving efficiency, by selecting numerical methods that are both fast and accurate, is more or less traditional. It was and is always used when large mathematical models (not only large LRTAP models) are to be handled on computers and, therefore, the basic principles are well-known (and more or less well-understood also). One attempts to improve the numerical methods selected for the model under consideration or even to develop special methods designed for the particular problems that are to be solved. However, it should be emphasized here that, while it is quite clear (in principle, at least) what has to be done, the actual task of selecting optimal (for the particular problem solved) numerical methods or of constructing new numerical algorithms which are especially designed for the problem under consideration is as a rule extremely difficult.

The benefits of a correct choice of numerical algorithms will be demonstrated in this chapter by only one example: by explaining the development of self-adaptive algorithms for solving the systems of ODE's that arise after the discretization of the advection part of an LRTAP model. This example will illustrate the fact that some solid theoretical results from the field of numerical analysis are needed in the efforts to design efficient numerical methods for large LRTAP models. The example will also explain the difficulties that arise when very big application packages (not necessarily large air pollution models) are to be handled efficiently on modern computers. The difficulties become greater when non-linear chemical reactions are included in the model. It should be emphasized here that while the benefits of using this approach in models with simple linear chemistry are obvious, the problem of designing the algorithm in the case where non-linear and photochemical reactions are involved in the model is still open. Of course, many other examples, where similar difficulties are to be overcome, can be given.

While the use of fast numerical algorithms in the big air pollution codes is necessary, as a rule this is by far not sufficient. The codes should normally be run on modern high-speed computers (and in the most cases the task of achieving high performance on such computers is also an extremely difficult task). The basic principles on which the high-speed computers are based will briefly be discussed in Chapter 6. The performance of some modules of **the Danish Eulerian Model** on three types of modern high-speed computers will be demonstrated in Chapter 7 - Chapter 9.

5.1. Size of the discretized models

It is worthwhile to emphasize, once again, the fact that the numerical problems, which are to be solved when regional air pollution models are handled on computers, **are very large**. This justifies the search for both fast and accurate numerical algorithms. On the other hand, it is almost clear that such algorithms will in general be rather complicated, because all simple algorithms have already been tried (many times and with numerous small modifications) by many authors.

Consider the air pollution model that has been discussed in Section 2.9. Assume that this model is split into five sub-models according to the physical processes that are involved in it (see the description given in Section 3.1). Assume also that the number of air pollutants which is to be studied by the model is **35**. Finally, assume that the space domain is discretized by using a $(32 \times 32 \times 10)$ grid. Then each of the five sub-models leads (after the discretization) to a system of **358400** ODE's that is to be treated numerically at each time-step. To complete the description, it must be stressed that the number of time-steps needed to complete the computations in a typical run of the model is usually rather large (several thousands if the time-interval is one month). If the calculations with such a model are to be performed:

(i) with a time-stepsize $\Delta t = 15$ min.,

(ii) on a time-interval of one month

and

(iii) with a start-up period of five days,

then the number of time-steps needed to complete a run with an air pollution model discretized as described above is **3456** (see, for example, Zlatev and Christensen [537], or Zlatev et al. [538,539]).

This simple example illustrates clearly that the computational work needed to treat an LRTAP model numerically is enormous. Unfortunately, this is not the whole story. The problem is that the chemical sub-model leads to systems of ODE's that are very stiff. The problem of stiffness is studied in detail in many books on ODE's (see, for example, Butcher [55], Gear [182], Hairer and Wanner [227] or Lambert [280,281]) and in connection with many well-known codes (see Butcher et al. [56], Gear [183,184], Hindmarsh [241] and Shampine [393,394,397,401]). In the particular case where large LRTAP models are treated numerically, the important issue is the fact that the stiffness of the chemical sub-model restricts the time-stepsize. As an illustration of this let us mention here that the chemical sub-model of **the Danish Eulerian Model** had to be run with a time-stepsize $\Delta t = 2.5$ min. instead of the time-stepsize $\Delta t = 15$ min. which is used in the other sub-models. This is a reduction by a factor of six. The small time-stepsize, $\Delta t = 2.5$ min., will be called the chemical stepsize, while the large time-stepsize, $\Delta t = 15$ min., will be called the advection stepsize. The fact that different stepsizes are to be used in the chemical

sub-model and in the other sub-models means that the actual number of systems of ODE's that are to be treated at each of the 3456 advection steps is **ten**. Thus, the total number of ODE's that are to be treated numerically at each advection step is **3.584*10⁶**.

In the newest long-range transport models the discretization is carried out by using finer grids; (96x96) grids are often recommended in the discretization of the horizontal grid-planes when the particular space domain of the model contains the whole continent of Europe. If such a discretization is used, then the number of ODE's, which are to be treated numerically at every advection time-step, will, roughly speaking, be increased by a factor of nine. It is difficult at present to run such large air pollution models on long time-intervals. This is especially true when the models are three-dimensional. However, models of this type could be used to study episodes (i.e. short periods with very high concentrations). It should also be emphasized here that in the near future it would probably be possible to run successfully three-dimensional models discretized on fine spatial grids over long time-intervals when new and fast computers are used.

The size of the problems that arise after the discretization of air pollution models can be reduced considerably (by a factor of ten in the particular case discussed in this section) when the models are considered as two-dimensional models as in Harrison et al. [228], Zlatev and Christensen [537] and Zlatev et al. [538-540]. This explains why many LRTAP models are still used as two-dimensional models.

For the sake of completeness, it must be emphasized here, once again, that the size of the discretized three-dimensional air pollution models is **not** the only limiting factor when the three-dimensional case is to be treated. The lack of high quality three-dimensional input data (both meteorological data and emission data) is very often the most important limitation. However, the situation is gradually improving. The computers become faster and faster, and there are good reasons to believe that also high quality three-dimensional input data will soon become easily available.

5.2. Efficient implementation of ODE algorithms

Consider the two-dimensional advection sub-model (which is one of the most important, and also one of the most difficult, parts of the computational work when a large air pollution model is handled numerically). In Section 3.1 it has been shown, see (3.4), that the advection sub-model of an LRTAP model can be written as

$$(5.1) \quad \partial C_s^{(1)} / \partial t = - \partial (u C_s^{(1)}) / \partial x - \partial (v C_s^{(1)}) / \partial y,$$

where $s = 1, 2, \dots, q$ and q is the number of pollutants, which are to be studied by the model.

Since we shall be interested only in the solution of advection part of the model in this section, it is not necessary to consider all q PDE's. The equations in the system defined

by (5.1) are independent and, therefore, it is quite sufficient, for our purposes here, to consider only one PDE (such a simplification has also been applied in Section 3.2). It will furthermore be assumed that the single equation can be written in the following form (see also Chapter 2):

$$(5.2) \quad \partial c / \partial t = - u \partial c / \partial x - v \partial c / \partial y,$$

where $x \in [0, 2\pi]$, $y \in [0, 2\pi]$ and $t \in [0, T]$.

Assume again, as in Section 3.2, that the following two conditions are satisfied:

(i) The space domain of (5.2) is discretized by an equidistant grid where $2M+2$ grid-points are used along each grid-line which is parallel to the Ox axis and $2N+2$ grid-points are used along each grid-line which is parallel to the Oy axis; see also (3.14) and (3.15).

(ii) the pseudospectral method (which is based on trigonometric interpolants which are truncated Fourier series) is to be applied in the space discretization of (5.2).

The discretization by the pseudospectral method leads, see (3.33), to the following system of ODE's (but it must be emphasized here that similar systems of ODE's can be obtained when many other discretization techniques are used):

$$(5.3) \quad dg/dt = - (US + VPS^*P) g.$$

The different quantities, which are involved in this system of ODE's, are fully described in Section 3.2. A brief description is given below.

(i) The vector $g = g(t)$ contains values of the concentrations at the grid-points (ordered by rows).

(ii) S and S^* are block diagonal matrices which are induced by the discretization rule used. The matrix S contains $2N+1$ diagonal blocks equal to the matrix S_M which has been defined by (3.29), while matrix S^* contains $2M+1$ diagonal blocks equal to matrix S_N which has been defined by (3.30).

(iii) U and V are diagonal matrices, the diagonal elements of which are values of wind velocities u and v on the grid-points.

(iv) P is a permutation matrix.

The ODE system (5.3) must be solved numerically. Moreover, it is important to choose numerical methods that could be used with as large time-stepsizes as possible. Assume that the space discretization gives sufficiently accurate results (or, in other words, that the errors caused by the space discretization are sufficiently small). It should be mentioned here that in practice it is very difficult to satisfy this requirement (the requirement for sufficiently accurate space discretization leads to a requirement of using very fine spatial grids). Let us, nevertheless, assume that the space discretization is sufficiently accurate. Then there are two major factors that impose restrictions on the choice of a time-stepsize (under the assumption that the space discretization is sufficiently accurate):

(i) the required accuracy of the results

and

(ii) the stability of the computations.

One can select special methods that will be stable for any stepsize (the so-called A-stable methods; see, for example, Butcher [55], Hairer and Wanner [227] or Lambert [280,281]). Such time-integration methods are very useful in some situations. A typical situation where this is true is the case where very stiff systems of ODE's are to be solved (the chemical sub-models lead to such systems and, thus, A-stable methods are a good choice in this case). However, one should be careful. The A-stable methods are implicit and, therefore, they lead to a solution of large systems of algebraic equations which is a very expensive procedure. The alternative possibility is the use of explicit methods. The explicit time-integration methods are much cheaper, but the stability requirements may restrict the time-stepsize. If the problem is not very stiff, then explicit methods can successfully be used. The advection sub-models do not lead to very stiff problems, and explicit integration methods with improved stability properties (as, for example, these proposed in Zlatev [512]) can successfully be applied in the numerical treatment of the systems of ODE's which arise after the discretization of these sub-models.

Assume that explicit time-integration methods are to be applied. Then very often the stability requirements (and not the accuracy requirements) impose a limiting restriction on the choice of the time-stepsize. Therefore it is essential to select explicit time-integration methods that have good stability properties. This means that some stability analysis is necessary before the discussion of the selection of particular time-integration methods, which are suitable for solution of the system of ODE's (5.3) when this system arises after the discretization of the advection sub-model (5.2). It is convenient to carry out the stability analysis for the simple test-equation (see Zlatev [512,517,518,521] and/or Zlatev et al. [529]):

$$(5.4) \quad \partial c / \partial t = -u \partial c / \partial x,$$

where the wind velocity u is assumed to be a constant.

The test-equation given by (5.4) is a PDE analog to the famous test-equation $dy/dt = -\lambda y$ that has been used in many studies concerning the stability of ODE systems.

Assume now that the pseudospectral discretization described in Section 3.2 is applied to (5.4). The result, which can formally be obtained from (3.33) by setting $V = 0$ and by assuming that U is a diagonal matrix whose diagonal elements are equal to u , is given by

$$(5.5) \quad dg/dt = -USg.$$

Assume that the time variable t is varied, as in (5.2), in the interval $[0, T]$, and consider an equidistant grid on the time interval:

$$(5.6) \quad \mathbf{t}_K = \{ t_k \mid t_0=0, t_{k+1}=t_k+\Delta t, \Delta t=T/K, k=0(1)K-1, K \in \mathbf{N} \},$$

where Δt is the time-stepsize (or, shorter, the stepsize).

Let g_k be an approximation to the exact solution $g(t_k)$. Assume also that all approximations up to some g_{k-1} have already been calculated. Then an explicit, forward stepping, time-integration algorithm for calculating the next approximation g_k is given by a formula of the following type:

$$(5.7) \quad g_k = \sum_{i=1}^s \alpha_i g_{k-i} + \Delta t \sum_{i=1}^s \beta_i f_{k-i},$$

where s is a given integer and

$$(5.8) \quad k = s, s+1, \dots, K, \quad f_j = -uSg_j, \quad j = 0, 1, \dots, K.$$

If $s=1$, then the method is self-starting; otherwise some starting procedure is needed to calculate some approximations $g_1, g_2, \dots, g_{s-1}$ of the concentrations at the first $s-1$ time-steps (the starting concentrations at the grid-points of the space domain, i.e. the components of vector g_0 , are to be given). Moreover, the starting procedure **may** be used (and in fact is often used) with some $\Delta t_1 \neq \Delta t$.

The time-integration method that is defined by (5.7) is called **the general linear multistep method** (see, for example, Butcher [55], Gear [182], Hairer et al. [226], Hairer and Wanner [227], Lambert [280,281], Lapidus and Seinfeld [283], Stetter [437]). Very often the following three algebraic polynomials

$$(5.9) \quad \rho(\xi) = \xi^s + \sum_{i=1}^s \alpha_i \xi^{s-i},$$

$$(5.10) \quad \sigma(\xi) = \sum_{i=1}^s \beta_i \xi^{s-i},$$

$$(5.11) \quad \pi(\xi, \bar{h}) = \rho(\xi) - \bar{h} \sigma(\xi),$$

are associated with the general linear multistep method (5.7) and used to derive many of the properties of these methods.

The algebraic polynomials defined by (5.9)-(5.11) are used in the three definitions given below. These definitions will be needed for the derivation of the main results concerning the absolute stability of the numerical time-integration algorithms.

Definition 5.1. It is said that an arbitrary algebraic polynomial $P(z)$ satisfies **the root criterion** if all roots of $P(z)$ lie on the unit disc (i.e. the disc defined by $|z| \leq 1 \wedge z \in \mathbb{C}$, where $\mathbb{C}$ is the complex plane) and if there are roots on the unit circle ($|z| = 1 \wedge z \in \mathbb{C}$), then these roots are simple. ■

Definition 5.2. Let $\bar{h}$ be an arbitrary complex number whose real part is non-positive (i.e. let $\bar{h} \in \mathbb{C} \wedge \text{Re}(\bar{h}) \leq 0$). The linear multistep method (5.7) is said to be **absolutely stable** for $\bar{h}$ if the third of its associated polynomials, the polynomial (5.11), satisfies the root criterion. ■

Definition 5.3. The set S_{abs} of all $\bar{h}$ with $\bar{h} \in \mathbb{C} \wedge \text{Re}(\bar{h}) \leq 0$ for which the linear multistep method (5.7) is absolutely stable is called **the absolute stability region** of the time-integration method. ■

The concept of **absolute stability** of a numerical method for integration of systems of ODE's has been introduced by Dahlquist [87]. The importance of this concept can be illustrated as follows. Let $u = 1$. Denote by λ_i , $i=1,2,\dots,2M+1$, the eigenvalues of matrix S from (5.5), which is induced by the discretization operator used (in our case, by the pseudospectral discretization algorithm); this matrix has been defined and studied in Section 3.2. Consider

$$(5.12) \quad \lambda = \max(|\lambda_i|), \quad i=1,2,\dots,2M+1, \quad \text{Re}(\lambda_i) \leq 0,$$

and

$$(5.13) \quad \bar{h} = \lambda \Delta t.$$

The linear multistep method defined by (5.7) will produce stable results when it is applied in the solution of (5.5) with $u = 1$ only if Δt is chosen so that h is inside the absolute stability region S_{abs} .

It is highly desirable (especially when large air pollution models are treated numerically) to use as large as possible time-stepsizes Δt , because the computing time is proportional to the number of time-steps. However, the above analysis shows clearly that both the space discretization method used and the linear multistep time-integration method selected impose some restrictions on the choice of the time-stepsize. The eigenvalues of matrix S , see (5.5), depend on the space discretization applied in the advection part of the air pollution models studied, while the absolute stability region S_{abs} depends on the particular linear multistep method chosen.

It will be shown in this section that some stability bounds for the time-stepsize Δt can be derived by exploiting some specific properties of the special problem under consideration (the advection parts of large air pollution models). These stability bounds can be used:

(i) to construct special linear multistep methods and predictor-corrector schemes for which the bounds are large

and

(ii) to apply the linear multistep methods and the predictor-corrector schemes so constructed with values of Δt which are close to the bounds found.

Some preliminary information concerning the eigenvalues of matrix S is needed. Such information is given in the following two lemmas. Lemma 5.1 is a simple modification (for our purposes) of a result obtained by Kreiss and Oliger [270]; see also Fornberg [169]. Lemma 5.2 is a simple corollary from Theorem 2.10.2 in Lancaster [282] (see also [180]).

Lemma 5.1. *Matrix S from (5.5) is anti-symmetric¹ and its eigenvalues form the set*

$$(5.14) \quad \Lambda = \{ \lambda_j \mid \lambda_j = (-M+j)i/j, \quad j=0,1,\dots,2M, \quad i^2=-1 \}. \quad \blacksquare$$

Lemma 5.2. *Any anti-symmetric matrix S is orthogonally similar to the diagonal matrix of its eigenvalues, i.e. there exists an orthogonal matrix H such that*

¹ The term "skew-symmetric matrix" is often used instead of "anti-symmetric matrix". A matrix is anti-symmetric if $a_{ij} = -a_{ji}$ holds for all of its elements.

$$(5.15) \quad S = H \Lambda H^T ,$$

where matrix Λ is a diagonal matrix the diagonal elements of which are the eigenvalues of matrix S . ■

A stability bound for the largest time-stepsize Δt that can be applied in the numerical integration of (5.5) by a linear multistep method can be obtained by using these two lemmas.

Theorem 5.1. Assume that

(i) h_{imag} is the length of the interval on the positive part of the imaginary axis (containing the origin) which belongs to the absolute stability region S_{abs} of the linear multistep method defined by (5.7)

and

(ii) the number of grid-points used in the space discretization is $2M+2$.

Then the application of the linear multistep method defined by (5.7) in the solution of (5.5) with any constant $u \neq 0$ leads to a stable computational process if the time-stepsize Δt is chosen so that

$$(5.16) \quad |u| M \Delta t < h_{\text{imag}} .$$

Proof. Use (i) the fact that the eigenvalues of matrix S are given by (5.14) from Lemma 5.1, (ii) the relationship (5.15) from Lemma 5.2, and (iii) the substitution $H^T g = z$ to split the system (5.5) into a set of $2M+1$ independent ODE's:

$$(5.17) \quad dz_\mu/dt = iu(-M+\mu)z_\mu, \quad \mu = 0, 1, \dots, 2M, \quad i = \sqrt{-1} .$$

The computational process that is carried out during the solution of any of the independent ODE's in (5.17) will be stable (see, for example, Dahlquist [87]) if $|u| |-M+\mu| \Delta t < h_{\text{imag}}$. Global stability, for the whole set of equations (5.17), will obviously be achieved by choosing the time-stepsize Δt so that (5.16) is satisfied (because $\lambda = \max(|-M+\mu|) = M$ for $\mu = 0, 1, \dots, 2M$). ■

The relationship (5.16) is derived under the assumption that $x \in [0, 2\pi]$. However this is not a restriction. If $x \in [a, b]$, then it is easy to derive the following relationship

$$(5.18) \quad \Delta t < [h_{\text{imag}} / (|u| M)] [(b-a) / (2\pi)] .$$

Both (5.16) and (5.18) are valid in the case where the number of grid-points used in the space discretization is even ($2M+2$ grid-points). This is not a restriction either. If the number of grid-points used in the pseudospectral discretization is odd (say, $2M+1$ grid-points), then a relationship for Δt can be obtained from (5.16) and (5.18) by replacing M with $M-1$. The bound for Δt that corresponds to (5.18) is given by

$$(5.19) \quad \Delta t < \{h_{\text{imag}} / [|u| (M-1)]\} [(b-a) / (2\pi)] .$$

The linear multistep method (5.7) and the time-stepsize Δt used in the numerical integration of (5.5) were discussed until now. However, the system of ODE's (5.5) is not a general system of ODE's. This system has several special properties arising from the fact that it is found after the pseudospectral discretization of the space derivative in (5.4) on the points of some grid of type (3.14). Therefore,

**it is necessary to relate the time-stepsize Δt
to the space increment Δx .**

Consider the case where the spatial grid contains $2M+1$ points. Then the relationship $\Delta x = (b-a)/2M$ holds, and the following stability bound is obtainable:

$$(5.20) \quad \Delta t < \{ [h_{\text{imag}} / (|u| \pi)] [M / (M-1)] \} \Delta x .$$

The bound of the time-stepsize that is given by (5.20) is only valid for the one-dimensional case; this is also true for the previous bounds (5.16) and (5.18). The following bound for the two-dimensional case is based on some heuristics (see also Zlatev et al. [529]). Assume that the two-dimensional problem (5.2) is defined for

$$(5.21) \quad x \in [a_1, b_1], \quad y \in [a_2, b_2] \quad \text{and} \quad t \in [0, T] .$$

Assume also that $\Delta x = \Delta y$ and $M = N$ (where $2M+1$ is the number of grid-points along the grid-lines parallel to the Ox axis, while $2N+1$ is the number of grid-points along the grid-lines parallel to the Oy axis). Let the quantities u^* and v^* be defined by

$$(5.22) \quad u^* = \max_{x \in [a_1, b_1], y \in [a_1, b_1], t \in [0, T]} (|u(x, y, t)|)$$

$$(5.23) \quad v^* = \max_{x \in [a_1, b_1], y \in [a_1, b_1], t \in [0, T]} (|v(x, y, t)|) .$$

Then the following bound for Δt will normally ensure stable computations when a linear multistep method (5.7) is used:

$$(5.24) \quad \Delta t < \{h_{\text{imag}} / [(u^* + v^*) \pi] [M / (M-1)]\} \Delta x .$$

Consider now the last bound for Δt . This relationship shows clearly that the length of the time-stepsize that could be used in the numerical integration with a formula of type (5.7) is limited by the following factors:

1.	the particular problems solved (through the quantities u^* and v^*),
2.	the pseudospectral discretization used (through both the number of grid-points and the space increment),
3.	the particular time-integration method chosen (through its stability interval on the positive part of the imaginary axis).

Of course, the accuracy that is required may also be a limiting factor for the time-stepsize Δt . This may be crucial for the choice of Δt in certain situations. Let us assume temporarily that the accuracy is not a limiting factor. Let us also assume that the spatial grid is fixed. It is clear that **if such an assumption is made, then an efficient time-integration algorithm is a time-integration algorithm that can produce stable results with as large values of h_{imag} as possible.**

It is well-known that there are some restrictions when integration algorithms of type (5.7) are to be used: the value of h_{imag} cannot exceed some bounds. These restrictions are discussed in Zlatev [512] and in Zlatev et al. [529]; the results in these two references are deduced from more general results in Jeltsch and Nevanlinna [259]. These results can be summarized in the following theorem.

Theorem 5.2. *Consider the class of linear multistep methods that is defined by (5.7). Let γ be an arbitrary real number in the interval $[0, 1]$ and let $s \in \{2, 3, 4\}$. Then there exists a particular linear multistep method of order $p=s$ in the class defined by (5.7) for which $h_{\text{imag}} = \gamma$. ■*

The upper bound (which can be achieved by taking $\gamma = 1$) can be obtained by the **leap-frog rule** (this is a second-order formula, which is often called the explicit mid-point rule). It is easily seen that the leap-frog rule can be found from (5.7) by setting $s=2$, $\alpha_1 = 0$, $\alpha_2 = 1$, $\beta_1 = 1$ and $\beta_2 = 0$. This means that the leap-frog rule is in some sense **optimal** when the class of linear multistep methods (5.7) is considered. However, the value $h_{\text{imag}} = 1$ is too small in many cases. Therefore it is often desirable to construct integration methods which can be used with a larger time-stepsize (this implies that for such methods the relation $h_{\text{imag}} > 1$ must be satisfied). It is necessary to consider a wider class of integration methods in order to be able to obtain methods with $h_{\text{imag}} > 1$. Such a wider class, **the class of predictor-corrector schemes with several different correctors**, can be introduced by following formulae.

$$(5.25) \quad g_k^{[0]} = \sum_{i=1}^{\mu^{[0]}} \alpha_i^{[0]} g_{k-i} + \Delta t \sum_{i=1}^{v^{[0]}} \beta_i^{[0]} f_{k-i},$$

$$(5.26) \quad f_k^{[0]} = (US + VPS^*P) g_k^{[0]},$$

$$(5.27) \quad g_k^{[r]} = \sum_{i=1}^{\mu^{[r]}} \alpha_i^{[r]} g_{k-i} + \Delta t \beta_0^{[r]} f_k^{[r-1]} + \Delta t \sum_{i=1}^{v^{[r]}} \beta_i^{[r]} f_{k-i},$$

$$(5.28) \quad f_k^{[r]} = (US + VPS^*P) g_k^{[r]},$$

where $r = 1, 2, \dots, q$; $q > 1$ being a given integer. It is assumed that

$$(5.29) \quad s = \max(\mu^{[0]}, v^{[0]}, \mu^{[1]}, v^{[1]}, \dots, \mu^{[q]}, v^{[q]}).$$

Then the formulae (5.25)-(5.28) are used for

$$(5.30) \quad k = s, s+1, \dots, K.$$

If $s=1$, then the scheme is self-starting. If $s>1$, then a starting procedure is needed to find some approximations to the solutions g_j and the right-hand side vectors $f_j = (US + VPS^*P) g_j$ for $j=1, 2, \dots, s-1$.

The method defined by the above formulae is called **a predictor-corrector scheme**. The practical implementation of the formulae (5.25)-(5.28) in the numerical treatment of a system of ODE's can be explained as follows. A prediction of the solution is calculated by using

(5.25). The prediction is used in formula (5.26) to evaluate the right-hand side of system (5.3). After that q corrections are computed by using (5.27). Each correction is followed by an evaluation, (5.28), of the right-hand side. After the last evaluation of the right-hand side of (5.3), one sets $g_k = g_k^{[q]}$ and $f_k = f_k^{[q]}$. Thus, everything is ready to start the next time-step; the time-step $k+1$.

Normally the same corrector formula is used in (5.27). The use of different corrector formulae provides more flexibility, which is important when particular predictor-corrector schemes with large h_{imag} are to be found.

The possibility of finding some special predictor-corrector schemes from the class (5.25)-(5.28) that can be applied with large time-stepsizes Δt is seen from the following theorem.

Theorem 5.3. *Consider the test-problem (5.4). The absolute stability interval on the positive part of the imaginary axis for a predictor-corrector scheme with q correctors from the class (5.25)-(5.28) is given by*

$$(5.31) \quad 0 \leq h_{\text{imag}} \leq q+1. \quad \blacksquare$$

Theorem 5.3 tells us that it is possible, in principle at least, to construct a predictor-corrector scheme with, say, three correctors whose absolute stability interval on the positive part of the imaginary axis is close to four. This is about four times greater than the maximal value that can be achieved when a single linear multistep formula (5.7) is used (see **Theorem 5.2**). However, the computational work per time-step is also increased by a factor of about four when a predictor corrector scheme with three correctors is used instead of a single linear multistep formulae. Furthermore, every formula in a predictor corrector scheme with three correctors (possibly different) is normally more complicated than the simple leap-frog rule for which the maximum absolute stability interval is achieved in the class of linear multistep formulae (5.7).

The above considerations are used by some authors to justify the statement that the leap-frog rule is in some sense optimal among the explicit integration schemes (the predictor-corrector schemes with a fixed number of correctors are usually considered as explicit methods). In general, this is true. However, in many big application codes there are other factors which should be taken into account. There are at least two reasons for using larger time-stepsizes in the treatment of the advection phenomena in a large air pollution model; even when the number of formulae needed to ensure stable computations with such large stepsizes is greater than one. **The first reason** is connected with the fact that some quantities (as, for example, the wind velocities) are to be calculated only once per time-step. **The second reason** is connected with the other physical phenomena (excluding the chemical part): if these can be treated with a large time-stepsize, and as a rule they can, then it is desirable to be able to apply a large stepsize also when the advection is handled. These and some other reasons are discussed in Zlatev [512].

There is another difficult problem that has to be solved when efficient predictor-corrector

schemes are to be used in large air pollution models. This problem is connected with the construction of schemes with large h_{imag} . **Theorem 5.3** is not a constructive theorem; it only states that predictor-corrector schemes with $h_{\text{imag}} \approx q+1$ (where q is the number of corrector formulae used) **may** exist. Such schemes are to be found in some way. In Zlatev [512] this has been done by using an optimization process (with the Harwell subroutine VA10AD; see Harwell Subroutine Library [229]). In this process

$$(5.32) \quad \mu^{[0]} = \mu^{[1]} = \dots = \mu^{[q]} = 2$$

was set (in the next section it will be explained why such a requirement is imposed). If the choice (5.32) is made, then the following relations are satisfied:

$$(5.33) \quad \alpha_2^{[r]} = 1 - \alpha_1^{[r]} \stackrel{\text{def}}{=} \alpha^{[r]}, \quad r=1, 2, \dots, q.$$

The parameters $\alpha^{[r]}$ ($r=0,1,\dots,q$) were used as free parameters (i.e. a set of parameters $\alpha^{[r]}$ was searched during the optimization process for which the predictor-corrector scheme has as large as possible h_{imag} subject to some restrictions on the error constants; see Zlatev [512]).

Some predictor-corrector scheme that have been obtained during the optimization process are given in Table 5.1; see again Zlatev [512]. Note that all predictor-corrector schemes listed in Table 5.1 are with different correctors. When different correctors are used, one has more free parameters in the optimization process. This leads, as a rule, to determining predictor-corrector schemes with longer absolute stability intervals on the positive parts on the imaginary axis. Plots of the absolute stability regions of the predictor-corrector methods found can also be drawn; some plots are given in Zlatev and Berkowicz [526].

5.3. Need for an automatic adjustment of the stepsize

It has been assumed until now that the accuracy requirements do not cause any trouble. This is, of course, not always true. Very often one have to take into account both the accuracy requirements and the stability requirements. This can be done by using the so-called **variable stepsize variable formula methods**. The variable stepsize variable formula methods allow us to adjust dynamically (according to the local requirements at the current time-step) the time-stepsize and, thus, to try to optimize further the computational process. These methods are especially very efficient when the wind velocity norms vary too much in the different parts of the time-integration interval. Therefore the variable stepsize variable formula methods are very useful in some situations and deserve some special attention. Variable stepsize variable formula methods based on predictor corrector schemes with large absolute stability intervals on the positive parts of the imaginary axis will be discussed in the following sections.

The space discretization of any large LRTAP model leads to the solution of several systems of ODE's (assuming here that the discretization is applied together with some splitting procedure; as that described in Chapter 3). The systems of ODE's are normally solved numerically by step-by-step time-integration algorithms. It is important to carry out the solution process in an efficient way. It has been explained in one of the previous sections, Section 5.2, that the code should try to select (at every time-step) as a large time-stepsize as possible. In this way the number of time-steps can be reduced and, therefore, this will lead (when correctly implemented, at least) to improving the efficiency of the computational process. The method is applicable for any computer and will also result in better performance on any computer, when it is correctly implemented.

Number of correctors	Type of the predictor-corrector scheme	Stability interval h_{imag}
1	(-0.0041,1.9884)-P ₁ EC ₃ E	1.995
	(-0.8440,1.8841)-P ₃ EC ₄ E	1.733
2	(-1.5909,0.3748,0.7710)-P ₂ EC ₂ EC ₃ E	2.330
	(-1.7631,0.5577,0.0044)-P ₂ EC ₂ EC ₃ E	2.462
	(-1.7631,0.5577,0.0044)-P ₂ EC ₃ EC ₄ E	2.842
3	(0.2885,0.9980,0.3446,1.0233)-P ₂ EC ₂ EC ₂ EC ₂ E	3.345
	(0.0375,0.3510,0.7124,0.0336)-P ₂ EC ₂ EC ₂ EC ₃ E	3.296
	(-2.1687-,0.2109,0.3695,0.2191)-P ₂ EC ₂ EC ₃ EC ₄ E	3.147

Table 5.1

Absolute stability intervals for some predictor-corrector schemes which may have several different correctors.

P_i is a predictor formula of order i.

E is an evaluation of the right-hand side.

C_i is a corrector formula of order i.

The figures in brackets are the values of the parameters α_r , $r=1,2,\dots,q$, found during the optimization procedure.

It is rather difficult to develop efficient air pollution codes, which will automatically try to select the largest possible time-stepsize at every time-step. The general purpose codes for the numerical solution of systems of ODE's are usually designed in such a way that the time-stepsize is changed by a small amount and, moreover, after each change of the time-stepsize several steps are carried out with the same stepsize; see, for example, Gear [182-184,186-188], Hindmarsh [240-242], Krogh [271-274], Petzold [351,352], Shampine [393-402], Shampine and Gordon [404]. This is not a good strategy when air pollution codes are handled; especially when the advection part of the model is handled (which is one of the most important and also one of the most time-consuming processes during the numerical treatment of large air pollution models). In the latter situation, i.e. for the important advection sub-models of large air pollution models, the time-stepsize depends strongly on the wind

velocities, which change, as a rule, rather rapidly. Therefore the variation of the time-stepsize in the algorithms by which the advection sub-model is treated should be designed so that during the whole integration process

the variation of the time-stepsize follows the variation of certain norms of the wind velocities in the space region of the model.

The solution of this important (and difficult) numerical problem will be discussed in next sections.

5.4. Variable stepsize variable formula methods

Consider again the advection part of a large air pollution model. Assume that the advection sub-model is discretized by a pseudospectral algorithm as described in Section 3.2. This is not a very restrictive assumption. Many of the ideas can also be applied when other discretization algorithms (as, for example, finite elements) are applied. However, it will be much easier to carry out the presentation when a particular space discretization is considered.

It has been shown (see Section 3.2 and Section 5.2) that the discretization by the pseudospectral algorithm on a $(2M+1) \times (2N+1)$ grid leads to the following system of ODE's:

$$(5.34) \quad dg/dt = - (US + VPS^*P) g .$$

The notation used in (5.34) is described in Section 3.2 and in Section 5.2. The description will not be repeated here.

It should be mentioned again that a system of ODE's of the same type as (5.34) can be obtained not only when the pseudospectral algorithm from Section 3.2 is used, but also with some other discretization rules (the use of any of the other discretizations, which lead to the same type of system of ODE's, will change only the matrices S and S^*). It should also be mentioned that many of the conclusions that will be derived in this section under the assumption that system (5.34) is obtained by the particular pseudospectral algorithm described in Section 3.2 are valid for other space discretization algorithms that lead to the systems of ODE's of the same type as (5.34). Some space discretizations (as, for example, some finite element methods) will lead to systems of ODE's that are not of the same type as (5.34). Nevertheless, as mentioned in the beginning of this section, many of the ideas can also be applied for such discretizations.

It has been shown in the previous section that predictor-corrector schemes with several different correctors can successfully be used in the numerical integration of the system of ODE's (5.34). It is desirable to use such a scheme with as a large time-stepsize Δt as possible. However, there is a restriction on the time-stepsize: it has been shown in Section 5.2, see (5.24), that one should expect the computational process to be stable when the relationship

$$(5.35) \quad \Delta t < \{h_{\text{imag}} / [(u^* + v^*) \pi]\} [M / (M-1)] \Delta x$$

is satisfied. This relationship is derived under the assumption that the same numbers of grid-points are used along the two horizontal axes and that the space increments are equal; i.e. $M = N$ and $\Delta x = \Delta y$. However, this assumption can easily be removed (see, Zlatev et al. [529]).

The quantity h_{imag} is called the absolute stability interval on the positive part of the imaginary axis (see again Section 5.2). The size of h_{imag} depends on the predictor-corrector scheme used. It has been shown in Section 5.2 how to choose the predictor-corrector schemes in order to get as large h_{imag} as possible. However, the direct use of (5.35) in the time-integration of a large systems of ODE's may (in some situations, at least) be very inefficient; even when h_{imag} is optimal. Therefore it is important to start the work towards the development of a procedure for an efficient stability control at every time-step with the solution of the following important task.

The main task, which is to be performed when both the space discretization method and the time-integration algorithm are already chosen, is to modify (5.35), so that the modified formula can easily be used to adjust locally the time-stepsize (trying to select as a large time-stepsize as possible at every time-step).

The quantities u^* and v^* are the maximal values of the wind velocity functions $u(x,y,t)$ and $v(x,y,t)$ that are defined as

$$(5.36) \quad u^* = \max_{x \in [a_1, b_1], y \in [a_2, b_2], t \in [0, T]} (|u(x, y, t)|),$$

$$(5.37) \quad v^* = \max_{x \in [a_1, b_1], y \in [a_2, b_2], t \in [0, T]} (|v(x, y, t)|).$$

Assume now that the space discretization scheme is fixed (i.e. $M = N$ and $\Delta x = \Delta y$ are given). Assume that also the time-integration scheme is fixed (with some absolute stability interval h_{imag}). Then the maximal time-stepsize that can be used depends only on the particular problem solved (through the quantities u^* and v^*).

In many treatments of LRTAP models, one is interested in running the model over a long time-integration interval, very often one year (in order to obtain some information about the seasonal variations of the concentrations; see, for example, Harrison et al. [228], Zlatev [515,517,520], Zlatev and Christensen [536,537] or Zlatev et al. [532-535,538-540]). If this is the case and if there are, say, two-three very stormy days, then the numbers u^* and/or v^* will be very large and the time-stepsize calculated by (5.35) will be very small. This situation is very unfortunate, because if the stormy days are removed, then it would be possible to carry out the time-integration with much larger time-stepsizes (and, thus, the number of time-steps could be reduced considerably). The example given here is very illustrative. It shows that in some cases it will be worthwhile to try to perform the computations with different time-stepsizes at the different parts of the time-interval. This can be done in the following way.

Assume that the time-integration process is completed up to some temporal point $t \in (0, T)$ and that it is desirable to calculate the concentrations at the next temporal point $t + \Delta t$ with as large Δt as possible. This means that at time t it is necessary to determine some optimal time-stepsize Δt . The relationship (5.35) can be relaxed if an optimal (or nearly optimal) time-stepsize has to be determined at every time-step. Let t^* and t^{**} be two real numbers that satisfy the relationships

$$(5.38) \quad 0 \leq t^* \leq t, \quad t + \Delta t \leq t^{**} \leq T.$$

Consider the quantities u^{**} and v^{**} instead of u^* and v^* respectively:

$$(5.39) \quad u^{**} = \max_{x \in [a_1, b_1], y \in [a_1, b_1], t \in [t^*, t^{**}]} (|u(x, y, t)|),$$

$$(5.40) \quad v^{**} = \max_{x \in [a_1, b_1], y \in [a_1, b_1], t \in [t^*, t^{**}]} (|v(x, y, t)|).$$

It is worthwhile, see also Zlatev [512] and Zlatev et al. [529,532], to replace the relationship (5.35) by

$$(5.41) \quad \Delta t < \{h_{\text{imag}} / [(u^{**} + v^{**})\pi] [M / (M-1)]\} \Delta x.$$

If this is done and if the interval $[t^*, t^{**}]$ is a sufficiently small neighbourhood of the current temporal point t (that contains $t + \Delta t$), then the bound for the time-stepsize that is given in (5.41) will be in many cases much less restrictive than that given in (5.35).

The use of (5.41) in the determination of the time-stepsize is not sufficient to ensure an efficient computational process. The problem is that, while one should expect a time-stepsize

calculated by (5.41) to ensure stable computations, such a stepsize may be too large and, therefore, the calculations with such a time-stepsize may lead to inaccurate results. This means that one has to keep the balance between the requirements for stable computations and the requirements for accurate results.

It is relatively cheap to obtain some estimation of the accuracy of the calculated solution when predictor-corrector schemes with several different correctors are used. Some norm of the difference between the solution calculated by the last corrector and the solution calculated by the predictor (or one of the previous correctors) gives normally a good estimation of the error in the approximate solution calculated by the last corrector.

The above discussion shows clearly that both the accuracy of the numerical solution and the absolute stability of the computations can efficiently be controlled at every time-step when predictor-corrector schemes are used in the numerical integration of system (5.34). Therefore it is possible (although this is a very difficult task) to develop a robust integration algorithm in which both the accuracy of the calculated solution and the absolute stability of the computational process are controlled in a reliable way, and (what is even more important), an attempt to perform the numerical integration of the advection sub-model with as large time-stepsizes as possible is carried out at the same time. The main ideas, on which such an algorithm is based, can be sketched as follows (more details can be found in Zlatev [512,517,521], Zlatev and Berkowicz [526] and Zlatev et al. [529,532]).

Consider an arbitrary time-step n . At the end of this time-step (i.e. when the calculations with the last corrector of the predictor-corrector scheme have already been performed) an error estimation of the accuracy of the calculated approximate solution is computed. The error estimation is used to accomplish two tasks:

- (i) To decide if the step has been successful (i.e. whether the approximate solution that has been calculated at the current time-step satisfies some prescribed accuracy requirements given by some error tolerance). If the step has not been successful, then it is rejected and the calculations are repeated with a smaller time-stepsize.
- (ii) To calculate (in the case where the step has been judged as successful) a time-stepsize that is as large as possible and, moreover, such that one should expect to obtain at the next step a sufficiently accurate approximate solution with this time stepsize (according to some prescribed error tolerance).

The device based on the above two rules is a well-known algorithm (and also a rather well-understood algorithm) that is commonly used in the development of modern self-adaptive ODE codes (see, for example, Gear [182-184,186-188], Hindmarsh [240-242], Krogh [271-274], Petzold [351,352], Shampine [393-402], Shampine and Gordon [404]). This is an excellent technique when the accuracy requirements are dominant over the absolute stability requirements (i.e. when only the accuracy is limiting the time-stepsize). If this is true, then the time-stepsize calculated at this stage (by using accuracy considerations only) will also ensure stable computations.

Even in the case where the accuracy is not the limiting factor for the time-stepsize of the integration algorithm, one can still use the above criterion (with some safety factor) to determine the time-stepsize. Moreover, the criterion is very often used in cases where the accuracy is not the limiting factor for the time-stepsize and the results are satisfactory. The reasons for this can be explained as follows. It can be shown, see Shampine [393,398,401], that if the absolute stability requirements are dominant and if the time-stepsize is determined by using accuracy requirements only, then a good time-integration code will produce sufficiently accurate results, but the calculations will normally be rather inefficient. Some price has to be paid when the absolute stability requirements are neglected. Assume that step n has been carried out by a time-stepsize that is sufficiently small, so that the calculations have been stable. Then the error is small and the next time-stepsize (calculated by the code will be large). This will as a rule lead to unstable computations. The absolute instability will manifest itself by producing a large error estimation (see again Shampine [393,398,401]). If large errors are obtained during the error estimation test, then this will imply either a rejection of the time-step (when the errors exceed the prescribed error-tolerance) or computations with a small stepsize at the next time-step (when the errors are large, but still under the prescribed error-tolerance).

There are many excellent general-purpose ODE codes, which are based on the direct use of (i) and (ii) (as, for example, the codes proposed in Gear [182-184,186-188], Hindmarsh [240-242], Krogh [271-274], Petzold [351,352], Shampine [393-402], Shampine and Gordon [404], Shampine and Watts [405]). The above analysis indicates that the codes based on (i) and (ii) will not always be very fast. It is also necessary to take into account the absolute stability requirements during the integration process. This is, in fact, done in some of the codes described in the above references. On the other hand, all these ODE codes are general-purpose codes, and it is very difficult to deal with the absolute stability in the general case. In the special case, however, where the advection part of a large LRTAP model is to be treated numerically, a much more efficient device based on (5.41) can be designed by exploiting the special properties of the particular problem treated. Assume that the calculations with (i) and (ii) have been used to determine a potentially optimal (with regard to the accuracy of the computations) time-stepsize at the end of a successful time-step n . Assume also that the calculations at the next time-step, step $n+1$, have been started and that the wind velocity field for step $n+1$ has been calculated. Then the tasks (i) and (ii) are to be followed by a third task:

(iii) Calculate the right-hand side of (5.41), and if the quantity obtained is smaller than the time-stepsize computed in (ii), then reduce the time-stepsize so that it becomes smaller than the right-hand side of (5.41); here some safety factor can be applied, see Zlatev et al. [532].

The device that is determined by (i)-(iii) takes into account both the accuracy requirements and the absolute stability requirements of the particular problem solved. Therefore it should work in a robust and efficient way when it is correctly implemented. However, it must be emphasized that it is not an easy task to implement this device correctly.

The time-stepsize can be varied from one step to another when the device described above is used. However, one can also vary the formulae in order to ensure greater flexibility. This leads to **variable stepsize variable formula methods (VSVFM's)** for numerical integration of large systems of ODE's. The first methods of this type have been devised in the end of 60'ies and in the beginning of 70'ies (see Gear [183,184], Krogh [272]). Now variable stepsize variable formula methods are often used in practice. However, it must be emphasized here that the variable stepsize variable formula methods are not always the best choice. In some cases the traditional methods, let us call them **constant stepsize constant formula methods (CSCFM's)**, may be a good choice.

One of the difficulties that are to be overcome when a VSVFM is to be implemented is connected to the so-called zero-stability. It is nearly trivial to ensure zero-stability when a CSCFM is used. The situation change (or, at least, may change) when VSVFM's are used, because the zero-stability properties of the predictor-corrector schemes may be lost during the transition from a CSCFM to a VSVFM (see, for example, Gear [186], Gear and Watanabe [191], or Zlatev [511,512,518,521]). This means that one should be careful when the set of formulae is selected and when a strategy for varying the time-stepsize is designed. The problems connected with the zero-stability of the time-integration methods will be treated in the next section. A formal definition of a variable stepsize variable formula method will also be given in the next section.

5.5. Zero-stability properties of the VSVFM's

A variable stepsize variable formula method, a VSVFM, can be based on a set $\mathbf{F} = \{F_1, F_2, \dots, F_m\}$ of m underlying predictor-corrector schemes. Of course, this is an example only; the VSVFM can be based on other integration methods. However, the basic ideas remain the same. The exposition of the main results can be facilitated by considering only predictor-corrector schemes.

The j 'th predictor-corrector scheme F_j ($j=1,2,\dots,m$) of the set $\mathbf{F}$ is given by

$$(5.42) \quad \mathbf{g}_k^{[0]} = \sum_{i=1}^{\mu_j^{[0]}} \alpha_{ji}^{[0]} \mathbf{g}_{k-i} + \Delta t \sum_{i=1}^{v_j^{[0]}} \beta_{ji}^{[0]} \mathbf{f}_{k-i},$$

$$(5.43) \quad \mathbf{f}_k^{[0]} = (\mathbf{US} + \mathbf{VPS}^* \mathbf{P}) \mathbf{g}_k^{[0]},$$

$$(5.44) \quad \mathbf{g}_k^{[r]} = \sum_{i=1}^{\mu_j^{[r]}} \alpha_{ji}^{[r]} \mathbf{g}_{k-i} + \Delta t \beta_{ji}^{[r]} \mathbf{f}_k^{[r-1]} + \Delta t \sum_{i=1}^{v_j^{[r]}} \beta_{ji}^{[r]} \mathbf{f}_{k-i},$$

$$(5.45) \quad \mathbf{f}_k^{[r]} = (\mathbf{US} + \mathbf{VPS}^* \mathbf{P}) \mathbf{g}_k^{[r]},$$

where $r = 1, 2, \dots, q_j$; $q_j > 1$ being some given integer. It is also assumed that the integer parameter s_j is determined by

$$(5.46) \quad s_j = \max(\mu_j^{[0]}, v_j^{[0]}, \mu_j^{[1]}, v_j^{[1]}, \dots, \mu_j^{[q]}, v_j^{[q]}).$$

It is clear that the formulae (5.42)-(5.45) that form the predictor-corrector scheme F_j can be used in the calculation of approximations to the solution of (5.34) when

$$(5.47) \quad k = s_j, s_j+1, \dots, K.$$

In many situations it is desirable to use the set of predictor-corrector schemes $\mathbf{F} = \{F_1, F_2, \dots, F_m\}$ in a VSVFM. This means that the equidistant time-discretization (5.6) from Section 5.2 should be replaced by a non-equidistant grid:

$$(5.48) \quad \mathbf{T}_K = \{t_k \mid t_0=0, t_{k+1}=t_k+\Delta t_k, \Delta t_k>0, k=0(1)K-1, K \in \mathbf{N}\},$$

where Δt_k is the time-stepsize at time-step k (the term *stepsize* will, as mentioned in the previous sections, be used when there is no danger for misunderstanding).

Assume that $s=s_j$ and consider the vector

$$(5.49) \quad \Delta t_{kj}^* = (\Delta t_{k-1}/\Delta t_k, \Delta t_{k-2}/\Delta t_k, \dots, \Delta t_{k-s+1}/\Delta t_k).$$

If the j 'th predictor-corrector scheme F_j from set $\mathbf{F}$ is to be used at time-step k and if the temporal grid is given by (5.48), then the coefficients in the formulae that are actually used in the computations will depend on the components of the vector Δt_{kj}^* (see, for example, Zlatev [512,521]). More precisely, in such a case the actual computations at time-step k will be carried out by the formulae:

$$(5.50) \quad g_k^{[0]} = \sum_{i=1}^{\mu_j^{[0]}} \alpha_{ji}^{[0]} (\Delta t_{kj}^*) g_{k-i} + \Delta t \sum_{i=1}^{v_j^{[0]}} \beta_{ji}^{[0]} (\Delta t_{kj}^*) f_{k-i},$$

$$(5.51) \quad f_k^{[0]} = (US + VPS^*P) g_k^{[0]},$$

$$(5.52) \quad \mu_j^{[r]} g_k^{[r]} = \sum_{i=1}^{\mu_j^{[r]}} \alpha_{ji}^{[r]} (\Delta t_{kj}^*) g_{k-i} + \Delta t_k \beta_{ji}^{[r]} (\Delta t_{kj}^*) f_k^{[r-1]}$$

$$+ \sum_{i=1}^{v_j^{[r]}} \Delta t_{k-i} \beta_{ji}^{[r]} (\Delta t_{kj}^*) f_{k-i},$$

$$(5.53) \quad f_k^{[r]} = (US + VPS^*P) g_k^{[r]}.$$

It should be stressed here that if the same time-stepsize has been used during the s_j last time-steps, then the formulae (5.50)-(5.53) are reduced to the underlying predictor-corrector scheme with constant coefficients; i.e. to the scheme given by (5.42)-(5.45). It should also be stressed that the order of the scheme (5.50)-(5.53) is the same as the order of the underlying predictor corrector scheme (5.42)-(5.45).

The concept of absolute stability was introduced and studied in the previous sections. There is a weaker concept, the concept of zero-stability, that becomes important when VSVFM's are used. Roughly speaking, a linear multistep method (when it is implemented as a CSCFM) leads to a stable computational process for a given Δt if it is absolutely stable. The absolute stability of a time-integration method is related to the distribution of the roots of the polynomials (5.11). Again roughly speaking, the zero-stability could be considered as a limiting case of the absolute stability, as absolute stability for $\Delta t = 0$.

This shows that the zero-stability properties of a linear multistep method (5.7) are related to the distribution of the roots of the polynomial (5.9). More precisely, this polynomial must satisfy the root criterion: all roots of this polynomial should be inside the unit disc and if there are roots on the unit circle these must be simple (see Definition 5.1 and Definition 5.2 in Section 5.2).

The statement made above is true under the assumption that single linear multistep formulae are used. However, the results can easily be extended for predictor-corrector schemes. In the latter case, it could be shown that the zero-stability properties of the whole predictor-corrector scheme depend on the zero-stability properties of its last corrector: if the last corrector of a predictor-corrector scheme is zero-stable, then the predictor-corrector scheme is also zero-stable; see Zlatev [521].

If CSCFM's (either single linear multistep formulae or predictor-corrector schemes with several correctors) are considered, then it is a rather easy matter to guarantee the zero-stability of the computations: one should only use formulae for which the polynomial (5.9) satisfies the root criterion, and it is rather easy to build such integration methods. However, the situation changes when VSVFM's are to be used. In the latter case, it is not sufficient to use zero-stable underlying formulae in order to ensure the zero-stability of the VSVFM based

on these underlying formulae. In other words, the zero-stability properties may be lost in the transition from CSCFM's to VSVFM's. An example where this happens is given in Gear and Watanabe [191]; see also Zlatev [511,512,518,521].

The problem of preservation of the zero-stability of the VSVFM's (such a problem does not exist at all when zero-stable CSCFM's are handled) must be solved in some way. There are many solutions of this problems. The simplest one is probably the use of the so-called two-ordinate underlying predictor-corrector schemes. A predictor-corrector scheme defined by (5.42)-(5.45) is a two-ordinate scheme if

$$(5.54) \quad \mu_j^{[r]} = 2, \quad r=0, 1, \dots, q.$$

If a predictor-corrector scheme is two-ordinate, then the requirement to have order of accuracy at least equal to one leads to

$$(5.55) \quad \alpha_{j2}^{[r]} = 1 - \alpha_{j1}^{[r]}, \quad r=0, 1, \dots, q$$

If $\alpha_{j1}^{[r]} = 1$, then $\alpha_{j2}^{[r]} = 0$ and the predictor-corrector scheme is in fact one-ordinate; such predictor-corrector schemes are normally called Adams schemes.

It is easily seen that any formulae (the predictor or any of the correctors) in a two-ordinate predictor-corrector scheme is zero-stable if $0 \leq \alpha_{j1}^{[r]} < 2$, because if this condition is satisfied, then all roots of the zero-stability polynomials derived from (5.9):

$$(5.56) \quad \rho_j(\xi) = \xi^2 - \alpha_{j1}^{[r]} \xi - (1 - \alpha_{j1}^{[r]}) , \quad r=0, 1, \dots, q,$$

are inside the unit disc (and if there are roots on the unit circle, then these are simple roots). In fact, the roots of (5.56) are $\xi_1 = 1$ and $\xi_2 = -1 + \alpha_{j1}^{[r]}$.

If all underlying predictor-corrector schemes (the schemes from set **F**) are two-ordinate, then any VSVFM which is based on them is also two-ordinate; i.e. its formulae satisfy relationships that are quite similar to those described by (5.54) and (5.55).

The use of a set of two-ordinate predictor-corrector schemes in the construction of VSVFM's is important, because it can be proved that the VSVFM's whose underlying predictor-corrector schemes are two-ordinate and zero-stable are also zero-stable under some very mild assumptions (i.e. the zero-stability properties of the underlying predictor-corrector schemes are preserved in the transition from CSCFM's to VSVFM's under some very mild assumptions). It is necessary to introduce the concept of a stable stepsize selection strategy in order to formulate a theorem concerning the zero-stability of two-ordinate VSVFM's.

Definition 5.4. Consider a VSVFM. Assume that the following two conditions are satisfied:

$$(5.57) \quad \Delta t = \max_{1 \leq k \leq K} (\Delta t_k), \quad \Delta t \cdot K \leq c < \infty,$$

$$(5.58) \quad 0 < \bar{\alpha} \leq \Delta t_k / \Delta t_{k-1} \leq \bar{\beta} < \infty \quad \text{for} \quad \forall k \in \{1, 2, \dots, K\}.$$

Then it is said that a stable stepsize selection strategy is used in the VSVFM under consideration. ■

It must be stressed that the second relationship in (5.57) is important: it is telling us that if $K \rightarrow \infty$, then $\Delta t_k \rightarrow 0$.

Theorem 5.4. *If a VSVFM is based on a set F of two-ordinate and zero-stable predictor-corrector schemes and if the discretization T_k from (5.48) is such that the stepsize selection strategy is stable, then the VSVFM is zero-stable when $0 \leq \alpha_{ji}^{(r)}(\Delta t_k) < 2$ for $\forall k \in \{1, 2, \dots, K\}$. ■*

It is not necessary to prove this theorem here. The proof of this theorem and some other results concerning the zero-stability properties of the VSVFM's can be found in Zlatev [504,507,511,512,518,521]. On the other hand, several remarks concerning this theorem and its applicability in several situations, which occur often in practice, are needed.

Remark 5.1. The conditions (5.57)-(5.58) are very mild and it is natural to require that these are satisfied (see, for example, Zlatev [511,512,518,521]). The requirement for two-ordinate predictor-corrector schemes could be relaxed. One could require only **the last corrector** to be two-ordinate (see the above references again). ■

Remark 5.2. If it is allowed to change the stepsize and/or the formula (in this remark a formula means a predictor-corrector scheme) at every time-step, then **Theorem 5.4** relaxed as explained in Remark 5.1 is in some sense optimal. This means that one can construct examples where zero-stable three-ordinate underlying predictor-corrector schemes (or, more precisely, predictor-corrector schemes in which the last correctors are three-ordinate formulae) lead to zero-unstable VSVFM's. Such an example is given in Zlatev [512]; this example is an extension of a simpler case (concerning single linear multistep formulae) that has been described in Gear and Watanabe [521]. ■

Remark 5.3. The previous remark shows that there is a **barrier** on the number of ordinates used in the last corrector if one wants to guarantee the zero-stability of a VSVFM in the case where both the stepsize and the formula (in this remark formula means predictor-corrector scheme) can be varied at each time-step: in such a situation the last corrector formula must be two-ordinate. However, the barrier can be overcome if the same stepsize and/or the same formulae are to be used successively during several time-steps. VSVFM's in which it is not allowed to vary the stepsize and/or the formula at every time-step have been studied by Cruzeix and Lisbona [80], Gear and Tu [190], Gear and Watanabe [191], Grigorieff [218], Grigorieff and Paes-Leme [219], März [311,312] and several other numerical analysts. ■

Remark 5.4. In situations where large air pollution models are treated one should probably allow both the stepsize and the formula to be varied at every time-step. Indeed, the absolute stability properties of the method depend strongly on some norms of the wind velocity vectors. The wind velocities vary as a rule rather quickly. This means that if the time-stepsize is not increased at a given time-step, there is no guarantee that one will still be able to increase it after several steps. The basic rule that has always to be applied in the choice of a time-stepsize during the numerical treatment of air pollution models can be defined as follows: **do not try to delay the increase of the time-stepsizes, increase it every time when you can.** ■

Remark 5.5. In some other situations it may be worthwhile to carry out the numerical integration process by using VSVFM's in which the same stepsize and/or the same formula is used during several time-steps. A typical situation where this is satisfied is the case where implicit integration methods are to be used (as, for example, in the chemical sub-model). In this situation one is interested to use the old Jacobian matrix as long as possible. Therefore the same stepsize and the same formula are to be kept as long as an old Jacobian matrix can be used. ■

5.6. Control of the accuracy of the results

Assume that a VSVFM that is based on a set $\mathbf{F}$ of predictor-corrector schemes is zero-stable (i.e. such that the conditions of **Theorem 5.4** are satisfied). Then the VSVFM can efficiently be implemented only when both an absolute stability control and an accuracy control can reliably be incorporated and used at every time-step. The absolute stability control can be based on the use of (5.41) when the pseudospectral algorithm is applied in the discretization of the first-order space derivatives. Some cheap accuracy control procedures that can be incorporated in a VSVFM will be discussed in this section.

Let TOL be an error tolerance (prescribed before the beginning of the computations). Let $g(t)$, $t \in [0, T]$, be the exact solution of (5.34). It is desirable to have $\|g(t_k) - g_k\| \leq \text{TOL}$ (or, in other words, the global error at the point $t_k \in \mathbf{T}_k$ should not exceed the error

tolerance). To be able to control the global error is the most desirable achievement, but this is a very ambitious task. Even when an estimation of the global error can be achieved, such an estimation is as a rule very expensive and, moreover, it is difficult to implement a global error estimator in a big application package. Therefore an estimation of the local truncation error is normally calculated and used in the error control at each integration step.

There are different ways of calculating an estimation of the local truncation error (see, for example, Butcher [55], Dekker and Verwer [99], Gear [182], Hairer et al. [226], Hairer and Wanner [227], van der Houwen [245], Lambert [280,281], Lapidus and Seinfeld [283], Shampine [402], Shampine and Gordon [405] or Stetter [435]).

If a predictor-corrector scheme is used, then a rather crude (but quite sufficient in many practical situations) estimation of the local truncation error of the time-integration algorithm can be achieved by taking the norm of the difference between the approximations obtained by two formulae of the predictor-corrector scheme.

Let g_k be, as usual, the approximation calculated by the last corrector. Let g_k^* be an approximation obtained by one of the other formula in the predictor-corrector scheme. Then an estimation, $EST_{k,j}$, of the local truncation error (made when F_j is the underlying predictor-corrector scheme of the predictor-corrector scheme that has been used at time-step k) can be found by

$$(5.59) \quad EST_{k,j} = \|g_k - g_k^*\|.$$

The problem of choosing the formula, which is to be used in the calculation of g_k^* , is easily solved when only one corrector is used; the predictor can be used in this situation. Assume that j_k different formulae are used at time-step k . Then the j_k-1 'st formula can be used to calculate g_k^* . Other choices of a formula (from the predictor-corrector scheme), by which g_k^* can be calculated, can also be used.

The case where the last corrector has order of accuracy higher than the order of accuracy of the formula, let us call it E_i , that has been used to calculate g_k^* is important. In this case the estimation given by (5.59) is in fact an estimation of the local truncation error made when E_i has been used. The final result, g_k , obtained by the last corrector will normally be more accurate (because it has been obtained by a more accurate formula). Thus, the error estimation $EST_{k,j}$ is rather conservative in this situation. On the other hand, since $EST_{k,j}$ is only an estimation of the local truncation error (not of the global error), it is clear that the error check will tend to be more robust in this situation. Therefore it is probably a good idea to try to achieve this situation; to construct underlying predictor-corrector schemes in which any of the last correctors has an order of accuracy that is higher than the order of accuracy of the other formulae involved in its predictor-corrector scheme.

5.7. Designing algorithms with desired properties

Consider the time-step k . An approximation g_k to the solution of (5.34) is found at the end of this step and the following test

$$(5.60) \quad \text{EST}_{k,j} < \text{TOL}$$

can be used in order to decide whether the step should be accepted or not. If the test (5.60) is not successful (i.e. if this inequality is not satisfied), then the time-step has to be rejected and the calculations to be repeated by a smaller time-stepsize (in many codes the computations are repeated with a time-stepsize that has been reduced with a factor of two when this happens).

If the test (5.60) is successful, i.e. if the inequality is satisfied, then the time-step is considered as a successful one. The next important task now is to calculate an optimal time-stepsize for the next time-step; the time-step $k+1$. As mentioned before, both the accuracy requirements and the restrictions imposed by the absolute stability of the numerical integration has to be taken into account.

Let us start with the accuracy requirements during the numerical integration process. Assume that $F_j \in \mathbf{F}$ is the underlying predictor-corrector scheme of the predictor-corrector scheme that has been used at step k . Assume also that the accuracy order of F_j is $p=p_j$ and that $\text{ACCUR}_{k+1,j}$ is some estimation of the accuracy that could be achieved at time-step $k+1$ if a predictor-corrector scheme corresponding to F_j will be used. Then it is well-known (see, for example, Gear [182-184], Hindmarsh [240-242], Krogh [271-274], Krogh and Stewart [275-276], Shampine and Gordon [405], Zlatev and Thomsen [545]) that a good guess for an optimal stepsize Δt_{k+1} for the next time-step (let us reiterate: under the assumption that a predictor-corrector scheme corresponding to F_j will be used) is given by the following expression for $\text{ACCUR}_{k+1,j}$:

$$(5.61) \quad \text{ACCUR}_{k+1,j} < \delta (\text{TOL}/\text{EST}_{k,j})^{1/p} \Delta t_k,$$

where $0 < \delta < 1$ is a safety factor ($\delta=0.9$ is commonly used).

It is not necessary, as mentioned in the previous sections, to use a predictor-corrector scheme corresponding to F_j (the predictor-corrector scheme which has been used at time-step k) also at time-step $k+1$. A predictor-corrector scheme that is corresponding to another underlying predictor-corrector scheme can be applied if the calculations with this other scheme can be carried out with a larger stepsize. Let us assume that $s \leq m$ similar expressions, $\text{ACCUR}_{k+1,i}$ ($i \in \{i_1, i_2, \dots, i_s\} \subseteq \{1, 2, \dots, m\}$), can cheaply be found for several other predictor-corrector schemes from the set $\mathbf{F}$. By cheaply calculated we mean, among other things, that for a given i the expression $\text{ACCUR}_{k+1,i}$ is calculated without calculating an approximation g_k by using a predictor-corrector scheme that corresponds to the

underlying scheme F_i (an approximation g_k is calculated only by using a scheme corresponding to F_j). Then the optimal time-stepsize for the next time-step $k+1$ is given by

$$(5.62) \quad h_{k+1} = \max_{i \in \{i_1, i_2, \dots, i_s\}} (\text{ACCUR}_{k+1, i}),$$

when only the accuracy requirements are taken into account. It should be mentioned here that if the maximal value in the right-hand side of (5.62) is achieved for some i which is not equal to j , then the choice made by (5.62) will also lead to a change of the underlying predictor-corrector scheme (from F_j to F_i).

If the accuracy requirements are dominant over the absolute stability requirements, then the choice made by (5.62) will also satisfy (5.41) and one can set $\Delta t_{k+1} = h_{k+1}$. Both the accuracy requirements and the absolute stability requirements will be satisfied in this situation. However, it is very often true that not the accuracy requirements, but the absolute stability requirements are dominant when the advection part of a large air pollution model is treated numerically. Therefore an absolute stability check is normally also needed. Such a check could be made in the beginning of time-step $k+1$; when the quantities u^{**} and v^{**} have already been calculated. More precisely, one can calculate the quantity:

$$(5.63) \quad h_{k+1}^* = \delta_1 \{h_{\text{imag}} / [(u^{**} + v^{**})\pi] [M / (M-1)]\} \Delta x,$$

where $0 < \delta_1 < 1$ is a safety factor which is introduced to guarantee that (5.41) is satisfied when h_k^* is inserted in its left-hand side instead of Δt ($\delta_1=0.9$ will normally be a good choice).

If both h_{k+1} and h_{k+1}^* are calculated, then it is easy to choose an optimal time-stepsize for the time-step $k+1$. One should expect both the accuracy requirements and the absolute stability requirements to be satisfied if the time-stepsize is chosen by

$$(5.64) \quad \Delta t_{k+1} = \min(h_{k+1}, h_{k+1}^*).$$

Consider two of the underlying predictor-corrector schemes, say $F_j \in \mathbf{F}$ and $F_i \in \mathbf{F}$. Assume that the following three conditions are satisfied:

- (i) The absolute stability interval along the positive part of the imaginary axis of F_j is greater than the absolute stability interval along the positive part of the imaginary axis of F_i .
- (ii) The accuracy order of F_i is either higher than the accuracy order of F_j or, at least, equal to the accuracy order of F_j , but in the latter case the error constant of the last corrector formula of F_i is less than that of F_j .

(so that one should expect the scheme F_i to produce more accurate results than F_j also in the case where both schemes have the same order of accuracy).

(iii) The number of formulae involved in F_i is less than the number of formulae involved in F_j (this means that if both predictor-corrector schemes are used with the same stepsize, then F_i is less expensive than F_j).

Assume furthermore that (i)-(iii) hold for any two of the predictor-corrector schemes in the set $\mathbf{F}$. Then the set $\mathbf{F}$ is called **optimal**. If an underlying predictor-corrector scheme F_j in an optimal set $\mathbf{F}$ has better absolute stability properties than another predictor-corrector scheme, then F_j is both less accurate and more expensive (because more formulae are involved in it). On the other hand, in many cases F_j can be used with a larger time-stepsize.

It is worthwhile to try to construct an **optimal** set $\mathbf{F}$ of underlying predictor-corrector methods. Assume that the set $\mathbf{F}$ is optimal. Assume also that

$$(5.65) \quad h_{k+1} > h_{k+1}^*.$$

In this situation the time-stepsize is limited by the absolute stability requirements. Therefore it is smaller than the stepsize that would be determined if only the accuracy of the results is taken into account. One can try to reduce the computational cost at step $k+1$ (that has been increased due to the fact that the stepsize has been reduced because of the absolute stability requirements) by switching to a predictor-corrector scheme in which fewer formulae are involved. The accuracy requirements will be satisfied when such a switch is made (because any of the underlying predictor-corrector schemes that involve fewer formulae is more accurate due to the definition of optimal sets $\mathbf{F}$). The computational cost will be reduced; simply because fewer formulae are involved in the computations.

The above discussion illustrates clearly that one should try to construct an optimal set $\mathbf{F}$ of underlying predictor-corrector schemes. A VSVFM based on such a set $\mathbf{F}$ will in general be more efficient than a VSVFM that is based on a set $\mathbf{F}$ which is not optimal. An optimal set $\mathbf{F}$ consisting of three predictor-corrector schemes has successfully been used in the treatment of some simple air pollution models (only two species and linear chemical reactions) in Zlatev [515,517,520], Zlatev and Berkowicz [526] and Zlatev et al. [532]. The predictor-corrector schemes from this set are given in Table 5.2. The orders of the formulae involved in these scheme is indicated in Table 5.2 by using indices. The figures that are given in brackets before the schemes are the free parameters $\alpha_j^{[r]}$ ($j=1,2,3$, $r=0,1,\dots,q_j$) used to obtain as large absolute stability intervals h_{imag} as possible; see also (5.33). It should be emphasized that any VSVFM, which is based on the optimal set of predictor-corrector schemes $\mathbf{F} = \{F_1, F_2, F_3\}$ described in Table 5.2, is zero-stable when the conditions of **Theorem 5.4** are satisfied.

Scheme	Type of the predictor-corrector scheme	h_{imag}
F_1	$(-0.3412, 0.3705, 0.5766, 0.4548) - P_2 EC_2 EC_2 EC_3 E$	3.26
F_2	$(0.6500, 1.5000, 1.0000) - P_2 EC_3 EC_4 E$	2.51
F_3	$(0.9000, 1.6000) - P_3 EC_4 E$	1.62

Table 5.2

Predictor-corrector schemes that form an optimal set F .

The indices show the orders of the formulae.

The figures in brackets are the values of the free parameters used in the construction of schemes with good absolute stability properties.

Option	Time-steps	CPU-time
CSCFM	1824	548
VSVFM	931	297

Table 5.3

Running the two options of the simple model on an IBM 3081 computer (the CPU-times are given in seconds).

The CSCFM was run with $\Delta t = 30$ min. (in this run this was a good choice, but in some other cases such a large time-stepsize may cause troubles), the average time-stepsize for the VSVFM is about 59 min.

Numerical results, which are obtained by using a VSVFM based on the set of predictor-corrector schemes from Table 5.2, are given in Zlatev [512,520,522], Zlatev and Berkowicz [526] and Zlatev et al. [532]. These results show that it is worthwhile to use a VSVFM when large air pollution models are to be treated numerically. Some results are given in Table 5.3 to illustrate the fact that a good VSVFM gives better results (in terms of computing time) than a CSCFM.

It is interesting to investigate the relationship between the norms of the wind velocity vectors and the time-stepsize when a VSVFM is used. This relationship is expressed analytically in (5.41). However, it is also illustrative to show how the norms of the wind velocity vectors are related to the time-sizes that are actually calculated during a computer run of the model. The variation of the norms of the wind velocity vectors and the variation of the time-sizes during a run of the simple model can easily be monitored during the computations. If this is done and if the ideas sketched above are correctly implemented, then it will clearly be seen that the time-sizes are small when the wind velocity norms are large and the time-sizes become large when the wind velocity norms are small. Plots

representing the variation of the time-stepsizes versus the variation of the norms of the wind velocity vectors are given in Zlatev [521], Zlatev and Berkowicz [526] and Zlatev et al. [532].

It must be emphasized here that the efficiency of the VSVFM's depends essentially on the implementation. If the implementation of a VSVFM has been successfully done, then the VSVFM will be efficient. However, it is not very easy to obtain an efficient implementation. For example, there are still some problems connected with the implementation of a VSVFM in the advanced models with non-linear chemical reactions (and this is the reason for running these models with constant stepsize constant formula methods; see Harrison et al. [228] and Zlatev et al. [538,539]).

5.8. Concluding remarks and references concerning the implementation of variable stepsize variable formula methods

The time-discretization of the advection part of any large LRTAP model leads to several huge systems of ODE's. The size of the numerical problems to be treated is enormous. Therefore it is important to apply efficient algorithms. These problems are discussed in **Section 5.1**.

It is crucial to reduce the number of time-steps in the numerical integration of this system of ODE's. The time-integration methods are to be chosen carefully. The stability of the computational process is one of the important issues that are to be taken into account when the time-integration algorithm is chosen. The stability properties of some time-integration algorithms are discussed in **Section 5.2**.

It is necessary to develop codes in which the time-stepsize is automatically adjusted by the code. This can be done by using a variable stepsize variable formula method (a VSVFM). Such methods have often been used after the end of the 60'ies (when the first such methods were developed in the end of 60's; see Gear [182-184] and Krogh [272]). In **Section 5.3** it is explained why it is desirable to use VSVFM's.

However, in most of the VSVFM's one tries to control only the accuracy. Since the systems of ODE's that arise from air pollution models are slightly stiff, it is also necessary to check the absolute stability during the calculations. Although a good VSVFM code will solve the problem even when only an accuracy control is incorporated in it, the solution will not be very efficient (see Shampine [393]). Therefore it is better to introduce an absolute stability check also. In general it is not very easy to develop an absolute stability control strategy. However, in the case where the system of ODE's arises after the time-discretization of the advection part of a LRTAP model, there is a surprisingly simple and, at the same time, very reliable device on which such a strategy can be based. This important issue, the need of a reliable absolute stability control, has been discussed in **Section 5.4**.

The development of a reliable absolute stability check is not the only difficulty, which has to be overcome when a VSVFM is used in the solution of systems of ODE's. There is a serious danger that the computations will not be zero-stable. If a constant stepsize constant formula method (a CSCFM) is used, then the zero-stability problems can be solved in a nearly trivial way. The problem is that even if all underlying formulae are zero-stable, the zero-stability may be lost in the transition from a CSCFM to a VSVFM (examples where this happens are given in Gear and Watanabe [191] and Zlatev [512,521]). This means that one should be careful when the underlying formulae are selected. The solution of this problem is discussed in **Section 5.5**. Stability properties are also studied in [59] [333], [334], [353], [455].

When the two important tasks (the check of the absolute stability and the selection of underlying formulae that ensure the transition to a zero-stable VSVFM) are solved, one should also try to obtain sufficiently accurate results. In other words, an accuracy control is necessary. This third task is nearly classical. In many codes based on VSVFM's one tries in fact to treat only this task; see Gear [183,184], Hull et al. [247], Krogh [271-274], Shampine and Gordon [405]. It is not very difficult, in principle at least, to treat this task when predictor-corrector schemes are used. The problem of finding some error estimates is discussed in detail in many studies; see, for example, Shampine and Gordon [405]. However, it should be mentioned that normally estimates of the local truncation error only (and even estimates of the principal part of the local truncation errors) are calculated. It is much more desirable to calculate estimations of the so called global error, but this task is also much more difficult. In this chapter the classical way is followed; i.e. some estimates of the local truncation error are discussed, see **Section 5.6**. The experience indicates that this is quite sufficient for the case at the present level of development of codes for treatment of air pollution models. However, the development in this field is very fast. Therefore, it will not be very surprising if global error estimations will be used in the near future.

The VSVFM's are rather complicated, but also rather powerful, tools in the treatment of large air pollution models. Several different tasks are to be solved at each time-step when such methods are used. One must

- (i) estimate the local truncation error,
- (ii) decide if the current step is acceptable,
- (iii) calculate an optimal (or, at least a good) time-stepsize for the next time-step,
- (iv) perform an absolute stability check

and

- (v) select the right formula (in our case, the right predictor-corrector scheme).

The treatment of these tasks in an adaptive code is discussed in **Section 5.7**. Since the systems of ODE's that result from the discretization of the advection part of a large air

pollution problem are normally mildly stiff, the fourth task is rather important. Fortunately, it is also relatively easy in this special situation to perform an absolute stability check. The experiments indicate also that the device is not only very simple, but also quite reliable (see the results in Section 5.7 and similar results in Zlatev [512,515,517,520,522], Zlatev and Berkowicz [526] and Zlatev et al. [532]). However, it must be emphasized here that in general this is not so; the devices for checking the absolute stability of the solution process for general systems of ODE's are rather complicated, and there is still insufficient evidence for the reliability of these complicated and expensive devices.

CHAPTER 6

COMPUTATIONS ON HIGH-SPEED COMPUTERS

"Although the fastest computers can execute millions of operations in one second, they are always too slow. This may seem a paradox, but the heart of the matter is: the bigger and better computers become, the larger are the problems scientists and engineers want to solve".

Arthur Jaffe: *"Ordering the universe: The role of mathematics"*, SIAM Review, **26**(1984), p. 478.

The computational problems that are to be solved when large LRTAP¹ models are treated numerically on computers are huge (this has been mentioned in several preceding chapters; see, for example, Section 5.1). Therefore it is critical to perform the whole computational work (or at least the most time-consuming parts of it) efficiently. In fact, in many cases the large air pollution models are not tractable at all unless the codes by which they are treated numerically are sufficiently fast. There are two ways to achieve the highly desired efficiency.

The first way is based on the selection of fast (but also sufficiently accurate) numerical algorithms and on a careful implementation of these algorithms in the computer programs that treat the air pollution models numerically. This topic has been discussed in the previous chapters. In Chapter 5 it has been illustrated (by using an example) that the development of fast and accurate numerical algorithms for large air pollution models may be a very complicated task. It has also been emphasized there, in Chapter 5, that in many cases the selection of fast and accurate numerical algorithms or the construction of new fast and accurate numerical algorithms will not ensure that the large air pollution models can be treated successfully on traditional scalar and/or sequential computers. Normally, it is necessary to run the large models on modern high-speed computers. However, if modern high-speed computers are applied in the treatment of large air pollution models, then there is a second way to achieve efficiency.

This second way, which has been used mainly during the last two decades, is connected with the new powerful vector and parallel computers. An important feature of these modern high-speed computers is their ability to perform many millions of floating point arithmetic operations per second (and even billions of floating point arithmetic operations per second). However, it must be emphasized here (and this must be well-understood) that high-speed performance is only a potential capability of the high-speed computers. The high-speed performance (the high-speed of computations) can as a rule be achieved only if the numerical methods that are applied in the model under consideration are carefully selected and, moreover, if the programming work is done thoroughly. Thus, the use of modern high-speed computers imposes some very strong requirements that must be satisfied by the developers of codes for large LRTAP models (at least when a fast computational process is to be achieved, and this is nearly always highly desirable).

One must be aware that a numerical method that has been used successfully on a traditional sequential and/or scalar computer (or even on another high-speed computer) **may**

¹ **LRTAP:** Long range transport of air pollution.

be unsuitable for the high-speed computer on which the runs are to be carried out. This implies that any change from one high-speed computer to another **may** lead to a requirement to replace several numerical methods with new ones in some parts of the code (at least, in the case where it is desirable to achieve a high performance of the code).

The basic principles on which high-speed computers are based will be discussed in this chapter. In the subsequent chapters the performance of three different types of high-speed computers will be demonstrated by using some modules of **the Danish Eulerian Model**.

6.1. Classification of high-speed computers

The improvement of the numerical algorithms (which are used in the numerical treatment of large air pollution models; several such algorithms were discussed in the previous chapters) may lead to a substantial reduction of the computing time that is needed when the codes are run on a given computer. However, the improvement of the numerical algorithms alone is normally not sufficient in the efforts to make the model tractable numerically on the computer under consideration, and to use it in long series of simulation experiments (as, for example, in experiments with different emission scenarios). It is also necessary to use a high-speed computer in the runs. The high-speed computers are able, if the programming is done carefully, to achieve many hundreds of MFLOPS². Therefore very big problems from different fields of science and engineering that could not be solved two decades ago can successfully be run on such computers.

It must be reiterated that **the requirement for a carefully and correctly done programming work is very essential**. As a rule, it is not an easy task to achieve speeds of computations that are close (in some sense) to the peak performance of the high-speed computer under consideration. In fact, this could very often be a very difficult task, especially when the scientific and engineering problems are very large (and the tasks that are to be solved after the discretization and the splitting of the large LRTAP models are certainly very large).

The degree of difficulties, which arise when large application packages are optimized for runs on high-speed computers, depends on the particular high-speed computer that is to be used in the calculations. The main types of high-speed computers will shortly be discussed in this section. After that an attempt to explain what kind of difficulties arise on the different types of computers will be made.

The high-speed computers, which are available at present, can, roughly speaking, be divided into four major groups. The different groups of high-speed computers are given in Table 6.1.

² **MFLOPS**: million floating point operations (additions or multiplications) per second.

It should be emphasized that the classification of the high-speed computers, which is given in Table 6.1, is **not** the only possible classification of existing high-speed computers. One can also apply other criteria and, thus, obtain other classifications³. However, the classification, which has been chosen here, provides a clear way to divide the high-speed computers into different groups. Moreover, it is quite sufficient for the purposes of this book (and this is even more important).

It should also be stressed that the development in this important area (i.e. the development of high-speed computer architectures) is very rapid. More and more new high-speed computers and architectures are appearing, and the newest computers are often hybrids that have properties of several of the major groups that are listed in Table 6.1. Nevertheless, it is important to describe the major groups of computers in some more detail. The knowledge about the main principles on which these groups are based will be useful when more complex hybrid computer architectures are studied. The main principles, on which the computers of the four major groups (that are listed in Table 6.1) are based, will be described in the following sections of this chapter. Some examples that illustrate the effect of using these computers will be given in the following three chapters.

Group	Names of the computers
1	Vector processors
2	Parallel computers with shared memory
3	Parallel computers with distributed memory
4	High-speed computers with more complicated architectures

Table 6.1
Classification of high-speed computers.

³ Classifications of the high-speed computers, which are based on the concepts of "**SIMD computers**" and "**MIMD computers**", are often used. SIMD is an abbreviation of "single instruction multiple data", while "multiple instruction multiple data" is abbreviated to MIMD. These concepts and other related concepts are not used in this book. However, it should be emphasized that they are very useful in many other situations.

6.2. Vector processors

The vector processors were the **first** high-speed computers. The CRAY machines were the first computers that reported computational speeds greater than 100 MFLOPS. The vector processors, and among them the newest CRAY models, are still among the fastest high-speed computers on the market.

The basic principles that are used to design different vector computers will be described in this section. The description of the basic principles on which the vector machines are based is simplified, and perhaps oversimplified, but the main ideas are clearly seen when all details (these which in the context of this book are not very important) are omitted.

Assume that a given arithmetic floating point operation, for example a multiplication of two real numbers, has to be performed on the components of two large vectors a and b that are stored in two long arrays A and B of length L . Assume also that the result is a vector c that has to be stored in array C of length L . In a scalar computer the first components a_1 and b_1 of the vectors a and b are taken from the memory and loaded to the arithmetic unit and multiplied there. The result, $c_1 = a_1 b_1$, is then stored in the first component of array C . This procedure is repeated for the second components, for the third components and so on. After L multiplications the whole vector operation is completed. Assume that the multiplication of two real numbers (together with two load operations from the computer memory and one store operation to the computer memory) can be performed in m time-units. Then the time needed to multiply the two vectors is

$$(6.1) \quad T_1 = Lm + d_1,$$

where d_1 is some overhead that depends on the particular computer used in the calculations.

The situation is changed substantially when a vector computer is used. Two major achievements are of fundamental importance for the vector machines. It has been found that any floating point arithmetic operation consists of several parts, for example m parts, that can be performed in m successive **elementary steps** by different parts of the arithmetic unit of a vector processor. Each elementary step can be performed in one unit time. The term "**one clock period**" is commonly used to denote the time needed to perform one elementary step on a given vector processor. The clock period depends on the particular vector processor. The clock period for a given vector processor can roughly speaking be used to measure its speed: if the clock period is small, then the vector processor under consideration is fast.

Moreover, it has also been discovered that the arithmetic units do not need to communicate directly with the computer memory; it is much more profitable if the arithmetic units have access to special **vector registers**. The vector registers contain a limited amount of data only. However the data that are in the vector registers can be moved very quickly to the arithmetic units.

Assume now that the arithmetic operation under consideration consists of m elementary steps that are to be performed successively. Consider again the example given above. Then a vectorized process (roughly speaking, on any vector processor) is achieved as follows. The first two components a_1 and b_1 are taken from the vector registers and put in the arithmetic unit. All components that are in the vector registers are moved forward, and if necessary the last positions on the vector registers are filled in. The first elementary step of the multiplication is performed, and the result is pushed forward in the arithmetic unit where the second elementary step is to be performed. At the same time, the next two components, a_2 and b_2 , are moved from the vector registers to the arithmetic unit to perform the first elementary step of the multiplication. Again, all components that are in the vector registers are moved forward, and if necessary the last positions on the vector registers are filled in. In the next step, the third one, a_1 and b_1 are pushed forward in the arithmetic unit (where the third elementary step of the multiplication will be performed), while a_2 and b_2 are pushed to the place where a_1 and b_1 have been before the third step (the second elementary step of the multiplication will be performed there). The next two components, a_3 and b_3 , are taken from the vector registers to the arithmetic unit to perform the first elementary step of the multiplication on them. The components of the two arrays in the vector registers are moved forward, and if necessary the vector registers are filled in. The process continues in this manner, a kind of **pipelining**.

It is clear that after m time units (clock periods), the multiplication of the first two components of the two vectors will be completed and the result can be loaded in array C . After that the computer will be ready (when certain conditions are satisfied) with one new component of vector c after every time unit (clock period). This means that the computing time (measured in clock periods) that is needed to complete the multiplication of the components of the two vectors is given by

$$(6.2) \quad T_2 = L + d_2,$$

where d_2 is again some overhead, which depends both on the particular vector processor used to carry out the computations and on the particular arithmetic operations that are performed in the loop under consideration. In the simple example considered here d_2 will depend linearly on m (i.e. d_2 will be of the form $m + d_3$, where d_3 depends on the computer used).

It is rather difficult to compare directly the computing times that are given by these two formulae, formulae (6.1) and (6.2), because the two overhead parameters d_1 and d_2 may differ very much (and, moreover, these overhead parameters will in general depend on the particular computers that are used in the computations).

It is quite clear, however, that computations performed on a given vector processor will be efficient (and the speed of computations will be high) if the parameter d_2 is small (compared with L). The efficiency achieved will, of course, depend strongly on the particular vector machine that is used. On the other hand, there are also several features that are typical for all vector processors; for any vector processor, the speed of computations will depend on:

- (i) the length of the vectors involved in the computations,
- (ii) the order of the components of the vectors in arrays where these vectors are stored,
- (iii) the flow of data from the memory to the vector registers and from the vector registers to the arithmetic units,

and

- (iv) some other factors (as, for example, on the presence of `if` statements in the loop that is executed by the vector processor).

If the vectors `a` and `b` are sufficiently long and if these are stored appropriately in the arrays `A` and `B`, then the important condition (iii) will be satisfied and the vectorization is successful (on most computers it is best to store vectors contiguously in the corresponding arrays).

It is useful to conclude the description of the basic vectorization principles with an example. Consider again the multiplication of the components of two vectors and assume that this operation has to be performed on a CRAY Y-MP processor. The numbers of elementary steps for the three operations that are needed when two components of the two vectors are multiplied are given in Table 6.2 (the figures given in Table 6.2 are taken from the recent CRAY manuals).

It should be mentioned that the figures that are given in Table 6.2 are derived for a rather special situation, the situation where the components of the vectors `a`, `b` and `c` are stored contiguously in the corresponding arrays `A`, `B` and `C`. This means that the calculations are to be carried out by the piece of code shown in Fig. 6.1. It is seen that any two successive components of the arrays `a`, `b` and `c` are stored in two successive positions of the corresponding arrays `A`, `B` and `C`. If this condition is not satisfied (i.e. if the components of the vectors are not stored contiguously in the corresponding arrays), then the figures in Table 6.2 will be different (and in general the speed of computations will be slower).

Operation	Number of elementary steps
Load	17
Multiply	7
Store	17

Table 6.2
The numbers of elementary steps needed to multiply two real numbers on a CRAY Y-MP computer.

```
DO 10 I=1,L  
  C(I)=A(I)*B(I)  
10 CONTINUE
```

Figure 6.1
Code for multiplications of the components of two vectors
when these are stored contiguously in two arrays.

Two important conclusions can immediately be drawn from the numbers that are shown in Table 6.2.

The first conclusion is that the manipulations with data (the load and store operations) would likely take about **five times** more computing time than the time needed to perform the actual floating point arithmetic operation (the multiplication of two real numbers).

The second conclusion is that there will be a delay of **40 clock periods** in the very beginning of the loop shown in Fig. 6.1. However, one result per clock period will be produced after that if certain requirements, some of them have been listed above, are satisfied.

There are many other issues that must be taken into account when an efficient code for vector machines has to be developed. Two of these issues are very important:

(i) the ability of many vector computers to **chain** certain arithmetic operations if some conditions are satisfied

and

(ii) the possibility to achieve a full exploitation (or a nearly full exploitation) of the vector registers when the computations are properly organized.

Chaining is a combination of several floating point arithmetic operations, such that the elementary steps continue from one arithmetic operation to another. This means that a process including more than one floating point operation is executed as one long floating point operation when chaining takes place. Therefore it is clear that chaining leads to a reduction of the overhead d_2 and increases the efficiency of the vectorized computational process. Unfortunately, on most of the vector machines the chaining process cannot be controlled directly by the programmer. On the other hand, the compilers chain automatically many of the relevant arithmetic operations. However, there are some principles that should be applied carefully in the programming work in order to enhance the chaining process during the computations.

The efficient use of the vector registers can also increase the speed of computations. Therefore it is often desirable to exploit fully the vector registers. In some situations a better exploitation of the vector registers can be achieved by using **unrolling of the inner loops**,

when the loop under consideration is a nested loop. The possibility of increasing the speed of computations by using unrolling of the inner loops of certain nested loops will be demonstrated in Chapter 7. The compilers of many vector machines try automatically to unroll some of the loops (and to produce a code where the loops are unrolled). The programmer can unroll some of the other loops and investigate the effect of this action. There are cases where the compiler is not able to perform the unrolling process properly. This is the case where some modifications in the code are needed in order to make the unrolling process possible. One example, in which nested loops (which arise in computations with banded matrices) are to be unrolled, will be given in the next chapter.

Modellers who are interested in this topic (vector processors and scientific computations on vector processors) could find more details in, for example, Ortega [339] and Schönauer [389].

6.3. Parallel computers with shared memory

As a rule, computers of this class contain only a few but tightly connected processors. The different processors can work simultaneously on different tasks. When several processors are working concurrently, they access the data from a global memory. It is said that the processors **"share the data"**. The major task is to identify the parts of the computational work that can be done in parallel and to ensure that the computer can run these parts concurrently (possibly by inserting some appropriate directives for the compiler). It may be necessary to synchronize some computations because the processors share the data.

Sometimes it will not be sufficient only to find the parts of a big application code that can be run in parallel. It may also be necessary to organize the computations so that the processors work with the same small amount of data as long as possible. This is the **data-locality principle**. The benefits of using the same data as long as possible can be explained as follows. Very often **a cache memory** is also used in parallel computers with shared memory. The cache memory is, as a rule, not very big. However, it is much faster to refer to data that is in cache than to data from the global memory of the computer. This is why one should try to apply algorithms in which the data that is in cache are used as long as possible. This means that the work must be organized so that **data from the cache memory is used mainly in the computations**. Moreover, the work must also be organized so that the contents of the cache are not changed too often. Assume that these two conditions are satisfied. Assume, furthermore, that all the processors are kept busy most of the time. Then the code will work very efficiently. The three conditions, which will ensure efficient performance on many parallel computers with shared memory:

1.	data from the cache must be used in the computations as much as possible,
2.	the contents of the cache should be changed as seldom as possible,
3.	all the processors should be kept busy as long as possible,

are important. However, it must be mentioned here that it is not always possible to satisfy these three conditions. Moreover, it is normally not possible to tell the compiler what amount of data must be stored in cache.

This short discussion explains why the task of obtaining a high level of performance is often a very difficult task. More details about data organization in parallel machines can be found, for example, in Wijsoff [481].

The processors in a parallel computer with shared memory are as a rule very powerful. Moreover, very often they have a vector capability. When this last condition is satisfied, the optimization of the computational process can be carried out on two levels:

- (i) one can try to vectorize the computations on every processor

and

- (ii) one can try to run the most time-consuming part of the computational work in parallel across the processors.

Parallel computers with shared memory are rather popular. This is mainly due to the fact that it is relatively easy to achieve a good performance when computers in this group are available. The compilers of these computers are normally very friendly to the user and give clear information about many of the actions that are to be taken in order to optimize the code. Nevertheless, a lot of work is as a rule needed when a big application code is to be optimized on such computers.

The processors of many parallel computers with shared memory, such as CRAY X-MP, CRAY Y-MP, CONVEX and ALLIANT, have vector capabilities.

6.4. Parallel computers with distributed memory

As a rule, parallel computers with distributed memory have more processors than parallel computers with shared memory. Each processor of a parallel computer with distributed memory has its own local memory. When all the processors (or some of them) work in

parallel, every processor uses its own data. Of course, the processors have to communicate in order to exchange data. It is necessary to connect the processors in some network in order to make the communications between them possible. There are different kinds of connections that are commonly used in parallel computers with distributed memory. The hypercubes are very popular. A brief, but adequate, description of the main principles, on which the hypercubes are based, can be found in Lustman et al. [298,299].

Assume that one processor, processor p_i , has to pass some message (for example, a set of data) to another processor p_j . The message is normally not sent directly from processor p_i to processor p_j (which could be done only if the two processors are neighbours in the net of processors). A path from processor p_i to processor p_j is used to deliver the message. The path will depend on the interconnection network. Even when the network is fixed, the same communication that is to be sent from processor p_i to processor p_j may be passed by using different paths (depending on the current situation in the processor network at the moment when the communication is to be sent). However, a path can always be defined by

$$(6.3) \quad i=i_0, i_1, \dots, i_{n-1}, i_n=j,$$

where n is the length of the path. If the processors p_i and p_j are neighbour processors, then $n=1$.

Communication between the processors of a parallel computer with distributed memory is as a rule an expensive process. Therefore the programmer must try

- (i) to reduce the number of communications as much as possible,
- (ii) to reduce, as much as possible, the amount of data that is to be passed in every communication between the involved processors

and

- (iii) to organize the computational process in such a way that all communications can be carried out by using the shortest possible paths.

Some of these three requirements may work (in some cases, at least) in opposite directions. For example, it is often possible to reduce the number of communications between processors only if the length of the communications is also increased. If this is the case, then the optimal solution of the above three tasks may depend on the particular parallel computer with distributed memory that is to be used. For some parallel computers with distributed memory it may be more profitable to send numerous but short messages, whereas for other computers of this kind sending a few long messages may be the better solution.

From this discussion it becomes clear that the first requirement is, of course, to determine the pieces of work that could be carried out in parallel. However, in many cases this will be

insufficient. From the above discussion it follows that it is also necessary to organize the computational process so that (a) the processors work as long as possible with their own data, (b) the communications (when these are needed) occur rarely and (c) the messages are short and have to be sent to neighbour processors only. It should be mentioned here that it is not easy to satisfy simultaneously all these conditions. In many cases this is simply not possible, and a compromise has to be found.

A special type of the parallel computers with distributed memory are the **massively parallel computers**. These computers may have many thousands of processors. For example, the Connection Machine CM-200 can be configured with up to $2^{16} = 65536$ bit-processors. Each processor of a massively parallel computer has its own local memory. The processors communicate with each other in the same way as other parallel computers with distributed memory. This implies that the same type of problems arise when a large scientific or engineering model is to be programmed and the resulting code is to be made efficient for any of the massively parallel computers. However, solving these problems is even more difficult than for ordinary parallel computers with distributed memory, because of the greater number of processors in a massively parallel computer.

If the choice of a massively parallel computer (or an ordinary parallel machine with distributed memory) is considered in connection with the treatment of a large scientific or engineering model, then one should be aware that the requirements (i)-(iii) are very restrictive and, therefore, these cannot be satisfied for any application problem. However, it must also be emphasized that there are many problems for which the use of big, massively parallel computers is very efficient. This is normally the case when large scientific and engineering models that are described mathematically by partial differential equations that are discretized on regular spatial grids. Many important application problems, which appear often in practice, satisfy these two conditions: i.e., they are described by partial differential equations and, moreover, they are discretized on regular spatial grids. Therefore, one should expect massively parallel computers to be rather efficient for such application problems when the programming work is carefully done. This is probably why the massively parallel computers are gaining more and more terrain in the field of large-scale scientific computing. This process will probably continue in the near future.

6.5. High-speed computers with more complicated architecture

Three groups of main principles on which high-speed computers can be based were described in the preceding three sections, Section 6.2 - Section 6.4. When these principles are taken into account, high-speed computers can be divided into three major classes:

- (1) vector processors,
- (2) parallel computers with shared memory

and

- (3) parallel computers with distributed memory.

However, it is possible to construct more complicated architectures, in which two groups or all three groups of the main principles are applied simultaneously, and many such architectures have been constructed. This tendency, to build more and more high-speed computers with complicated architecture, is clearly seen during the last several years.

One simple example of computers based on more complicated architecture was explained in Section 6.3; that is, the ideas of building single vector processors and parallel computers with shared memory can be unified. The result is **a parallel computer with shared memory each processor of which has a vector capability**. This approach guarantees, as has been described in Section 6.3, a greater flexibility: one has a possibility to optimize the code on two levels:

- (i) to vectorize on every processor

and

- (ii) to parallelize across the processors.

One can also unify all three principles on which high-speed computers are based. One example is the CEDAR computer, which has been developed at the Center for Supercomputing Research and Development at the University of Illinois at Urbana-Champaign. A good description of the main principles used in the construction of the CEDAR computer can be found in Kuck et al. [277].

The CEDAR computer consists of several clusters of processors (at present it can be configured with up to four clusters). Each cluster of processors has its own local memory (together with its own cache memory, which is much smaller than the local cluster memory, but access to the local cluster cache memory is very fast) and could work in parallel with the other clusters of processors. This means that at this level the CEDAR computer has some properties of a parallel computer with distributed memory. However, even at this level, CEDAR is not a pure parallel computer with distributed memory, because there is also a common global memory that could be shared by the different clusters (the access to the global memory being much slower than that to the local memory).

Each cluster of the CEDAR computer contains several processors (at present, up to eight processors). The processors in each cluster share the local cluster memory and can work concurrently. Thus, within each cluster of processors the CEDAR computer works as a parallel computer with shared memory.

Finally, each processor of the CEDAR computer is rather powerful and has a vector capability. This means that at the lowest level, at a processor level, the CEDAR computer is a vector machine.

Some details concerning the CEDAR computer are omitted here. For example, there is some degree of flexibility in the configuration of the machine. As an illustration of this statement, it should be mentioned that one can also configure CEDAR as a plain parallel computer with 32 processors (instead of configuring it as a computer with four clusters with eight processors per cluster).

From the above description it is clear that the programmer can try to achieve efficiency by optimizing the code on three levels:

(i) one can try to vectorize at a processor level, i.e., to carry out vectorized computations at every processors,

(ii) one can try to achieve parallelism across the processors at a cluster level, i.e., to carry out parallel computations within each cluster of processors

and

(iii) one can try to achieve concurrency across the clusters, i.e., to distribute some computations to the different clusters and to perform them in parallel).

Of course, it is not possible to achieve optimization on all these levels for any application problem. However, the CEDAR architecture makes it possible (in principle, at least) to exploit the positive properties of all three groups of main principles on which high-speed computers are based. This fact, together with the fact that there is a great degree of flexibility in the configuration of the machine, shows that the CEDAR approach is a very interesting way to build flexible, high-speed computers. Some of the main ideas applied in the development of the CEDAR architecture are already used by some computer companies in the development of their new products.

All high-speed computers that have been considered until now are based on the idea that

all their processors are of the same type.

This is not a necessary condition that must always be satisfied when high-speed computers are to be constructed. One can also connect (probably with some loose connections) different computers. Some experiments in this direction have been carried out. It is possible that in the future high-speed computers connecting heterogeneous processors will play an important role in the field of large-scale scientific and engineering computing. Heterogeneous networks of computers ranging from work-stations to supercomputers are gradually becoming commonplace in high-performance computing. For more information about heterogeneous computing see Freund and Conwell [176], or Khokhar et al. [264].

Until recently, each computer in such a network remained a separate unit, but now the Parallel Virtual Machine (PVM) package is available at many sites and can be used to develop truly heterogeneous programs for multiple computer systems. The PVM package is a software package developed at Oak Ridge National Laboratory and Emory University. It enables a heterogeneous collection of UNIX computers that are linked in a network to function as a single parallel computer. PVM applications can be run across a wide variety of architectures. PVM handles automatically all message conversions required when the linked computers use different data representations. Participating computers can be distributed anywhere in the world and linked by a variety of networks. It should be possible to distribute the tasks in an optimal way: to send to each computer in the network tasks that are most suitable to its architecture.

PVM is a public domain code. The PVM source code and user's guide can be obtained by electronic mail. The source code has been tested on many different computers. Visualization and debugging facilities are available in the PVM program. More details can be found in Beguelin et al. [26].

6.6. High-speed computers and large air pollution problems

The numerical problems that arise after the discretization of large air pollution problems are enormous (see Section 5.1). However, these are normally discretized on regular spatial grids. Moreover, the splitting procedures (which are commonly used in the numerical treatment of the models) lead to some natural parallelism. For example, the advection sub-models (one per each of the species involved in the air pollution model) can be run concurrently after the performance of some splitting of the air pollution model. Moreover, most of the final tasks, which are to be treated during the discretization process and which are as a rule among the most time-consuming tasks, are well-known (and relatively simple) basic linear algebra operations. The efficient implementation of basic linear algebra operations is more or less a well-understood task for all types of the high-speed computers that were described in the previous sections of this chapter. This shows that the large air pollution models are potentially very suitable for efficient runs on different types of high-speed computers. However, it is also important to emphasize, once again, that this does not imply that the optimization of the big codes, which are obtained after the discretization of large air pollution models (the discretization being carried out together with some splitting procedure), is an easy task. It has already been mentioned several times, both in this chapter and in the previous chapters, that in fact the opposite is true, because the numerical problems arising when large air pollution problems are treated on computers are very large.

The task of optimizing a large air pollution model on a high-speed parallel computer with distributed memory is normally a much more difficult task than the task of optimizing such a model of the other types of parallel computers (vector processors and parallel machines with shared memory). This is the case because in such a model one operates with very large sets of data which have to be distributed properly among the processors of the computer under consideration.

The use of single vector processors seems to be easiest. However, also in this situation many technical problems are to be solved. Nevertheless, it is as a rule not very difficult to obtain good efficiency when single vector processors are used.

The use of parallel computers with shared memory machines, and also some more complicated architecture like the CEDAR computer, is also straightforward; at least in principle. However, some complications may arise. A typical situation where difficulties may arise is the case where the computer has a cache memory, and the cache memory is to be exploited efficiently.

The use of parallel computers with distributed memory may, as already mentioned, be rather complicated; an optimal distribution of the data that is used in the model to the different processors of the parallel computer with distributed memory is as a rule a tough task.

The task becomes even more difficult when a massively parallel computer is to be applied (because the number of processors in such a computer is very large). On the other hand, however, the large air pollution models contain, as already mentioned above, a lot of natural parallelism. The natural parallelism arises mainly because the systems of PDE's, by which the large air pollution models are described mathematically, are as a rule discretized on regular spatial grids. Therefore, it should be possible to obtain a good performance also when massively parallel machines are used. Unfortunately, it is normally very difficult to obtain the right solution of the task of the optimization of a large air pollution model on a massively parallel computer (even though one knows that such a solution must exist).

At present it is still difficult to judge the suitability of parallel computers with more complicated architecture for numerical handling of large air pollution models. More efforts are needed in this direction in order to determine whether such computers are suitable for large air pollution models. However, it is quite clear that such computers are potentially very promising. This is especially true for the parallel computers with heterogeneous processors which could be a very powerful tool in the efforts to handle large air pollution models efficiently and to achieve optimal speeds of computations in the different parts of such a model.

6.7. Concluding remarks and references concerning the efficiency of the numerical algorithms and the use of high-speed computers

The use of efficient numerical methods is absolutely necessary, but the use of efficient numerical methods will normally not solve all problems that arise when a large LRTAP model is to be handled. It is also necessary to apply some high-speed computer. High-speed computers are capable of carrying out many millions of simple floating point arithmetic operations (additions and multiplications) per second. However, careful programming work has to be done in order to utilize completely the great potential power of these machines. The

principles on which the main types of high-speed computers are based were discussed in **Section 6.1**. Different types of high-speed computers as well as many applications of scientific computations on high-speed computers are studied, for example, in [26], [176]-[179], [264], [277], [339], [389]. More details about the high-speed computers as well as some other classifications of the high-speed computers can be found there (see also the references given in Ortega et al. [340]).

Vector processors are the oldest high-speed computers. The main principles on which the vector processors are built are more or less well-understood now, and it is not very difficult to achieve high-speed computations when such computers are used. Efficient kernels for performing some basic operations that appear again and again in the computations have been developed for many such computers; see, for example, [114], [115], [339], [389]. However, some difficulties may arise when very big application codes are to be optimized. A short description of the vector processors is given in **Section 6.2**. Some applications related to large air pollution models will be discussed in more detail in Chapter 7.

Parallel computers with shared memory are also popular among the scientists and engineers that work with large scale problems and have been used to optimize many big scientific and engineering codes. The basic principles on which these parallel machines are built are briefly discussed in **Section 6.3**. It is underlined there that the most important case is the use of parallel machines with shared memory whose processors have vector capabilities. Such parallel machines are more flexible: one can try to optimize the code on two levels (to achieve vectorized computations on every processor and to achieve parallel computations across the processors). More details about the parallel computers with shared memory and/or the application of such computers in the solution of some numerical problems can be found in [117], [177]-[179], [339]. The efficiency that can be achieved when modules from air pollution codes are run on parallel computers with shared memory will be discussed in Chapter 8.

The main principles on which parallel computers with distributed memory are based are described in **Section 6.4**. It is emphasized there that the programming work is more difficult when these computers are used than the programming work needed when the previous two types of high-speed computers are used. This is especially true for the so-called massively parallel machines where the number of processors can be very large (for example, up to 65536 processors for the CM-200). These machines are potentially very promising for problems that are in some sense regular (for example, systems of PDE's that are discretized on regular spatial grids). If the programming work is done carefully, then one should expect good results for such problems. A good description of different parallel machines with distributed memory (including here some descriptions of massively parallel computers) as well as a good discussion of the organization of the computations on parallel computers with distributed memory can be found in [179], [290], [298], [299], [339]. However, it is highly desirable to have a monograph with a concise description of the most important principles which are common for all such computers and which can easily be understood by scientists and engineers working with big application problems. Such a monograph is not available yet. The performance of some parts of **the Danish Eulerian Model** (which is described and tested in Zlatev et al. [538,539]) on several massively parallel computers will be discussed in Chapter 9.

During the past five years, more complicated computer architectures have been built that possess properties of different kinds of high-speed computers:

- (i) abilities to perform computations in a vector mode,
- (ii) abilities to use shared memory banks,
- (iii) abilities to distribute some data to different processors and/or clusters of processors

and/or

- (iv) possibilities to be configured with a relatively large number of processors (not necessarily the same type of processors).

This process will probably continue. The computers of this type have been discussed in **Section 6.5** by using a simple but illustrative example; the CEDAR computer, see Kuck et al. [277]. The heterogeneous parallel computers are an interesting group among these computers. The development in this area is very quick, and it is difficult to predict what the situation will be after several years. Moreover, until now many of the quite new architectures have neither been described in a proper way nor tested on sufficiently many big application codes. Therefore one should be prepared to perform a lot of programming work if a computer from this group is chosen. Some interesting references on this topic are [26], [176], [264], [443].

The problems connected with the application of different high-speed computers to run big air pollution codes are discussed in **Section 6.6**. It is emphasized that the simple move from a traditional computer to a high-speed computer is not sufficient to ensure high-speed computations. A careful programming work has to be done. Moreover, some of the numerical algorithms (which were successfully used when the code was run on a traditional computer) **may** be unsuitable for the high-speed computer chosen. This implies that it may be necessary to replace some of the numerical methods used on the traditional computers with new numerical methods that are suitable for the high-speed computer chosen. This means that the task of achieving high speed computations is normally very difficult. On the other hand, some very big air pollution models can be made tractable numerically only if (i) a high-speed computer is used in the runs and (ii) the programming work is careful and accurate.

CHAPTER 7

RUNNING AIR POLLUTION MODELS ON VECTOR MACHINES

"The architecture that first caused a widespread and substantial algorithm redesign activity in numerical computing is the vector processor".

K. A. Gallivan, R. J. Plemmons and A. H. Sameh: *"Parallel Algorithms for Dense Linear Algebra Computations"*, SIAM Rev., **32**(1990), 54-135.

The long-range transport air pollution models (the LRTAP models) are, as shown in the preceding chapters, extremely complex and, therefore, a great amount of computing time is required to run them on computers. This (together with the fact that the big vector processors are still among the fastest computers that are available at the market) explains why large air pollution models are often run on big vector processors (as, for example, different types of CRAY computers).

Vector machines are potentially very powerful. However, it is necessary to program the codes carefully in order to exploit fully the great power of these machines. This is not always an easy process, especially when very large application codes are to be designed for efficient runs on vector machines. Therefore it is important to analyze the codes carefully in order to determine which parts of the codes are most time-consuming and then to concentrate the attention first and foremost on these parts.

Normally, the most time-consuming parts of any large long-range transport air pollution model are modules that perform standard algebraic operations, such as matrix-vector multiplications, inner products, solving systems of linear algebraic equations and Fast Fourier Transforms. Most of these operations involve matrix calculations. **An important property of the matrices involved in these standard algebraic operations is (in many applications, at least) their bandedness.** The fact that the matrices, which are to be treated in large long-range transport air pollution models, are usually banded matrices is important, and efficient codes can be developed when the bandedness is properly exploited when matrix operations are performed. In this chapter it will be shown how the standard algebraic operations can be vectorized and run efficiently on vector computers.

Of course, one has to perform such standard algebraic operations not only in connection with the numerical treatment of large air pollution models, but also in the numerical treatment of many other large-scale problems that arise in different fields of science and engineering. Therefore, some parts of this chapter may be useful also for scientists who are working in other fields within the big area of large-scale scientific computation.

7.1. Separating the most time-consuming operations

Consider the system of PDE's that is commonly used, see Section 2.9 and Section 3.1, to describe mathematically the long-range transport air pollution models (the LRTAP models):

$$\begin{aligned}
 (7.1) \quad \partial c_s / \partial t = & - \partial (u c_s) / \partial x - \partial (v c_s) / \partial y - \partial (w c_s) / \partial z \\
 & + \partial (K_x \partial c_s / \partial x) / \partial x + \partial (K_y \partial c_s / \partial y) / \partial y \\
 & + \partial (K_z \partial c_s / \partial z) / \partial z \\
 & - (k_{1s} + k_{2s}) c_s(x, y, z, t) \\
 & + E_s(x, y, z, t) + Q_s(c_1, c_2, \dots, c_q), \\
 s = & 1, 2, \dots, q,
 \end{aligned}$$

where q is the number of pollutants that are involved in the model (the different variables involved in the model have been discussed in Chapter 2 and Chapter 3).

Let us consider the case where the model is two-dimensional. Assume that such a model is split into several sub-models according to the physical processes involved in it (see Section 3.1). It has been shown (Section 3.1) that it is appropriate to split the model into four sub-models: advection sub-model, diffusion sub-model, deposition sub-model and chemistry sub-model (the emission terms being normally added to the chemistry sub-model). If the model is three-dimensional, then special care should be taken when the computations in the vertical directions are carried out. However, the computations in the vertical directions could also be added to the four sub-models that are obtained by splitting according to the physical processes.

Consider first the advection sub-model. The discretization of the space derivatives of the advection sub-model leads to application of Fast Fourier Transforms (FFT's) if a pseudospectral algorithm is used. If finite differences are used and if implicit integration methods are applied to solve the obtained linear systems of ODE's, then one has to solve systems of linear algebraic equations whose matrices are banded. If finite elements are used, then one has to solve systems of linear algebraic equations whose matrices are banded even when explicit integration methods are used.

The banded matrices have a special structure: (i) many of their elements are equal to zero and, moreover, (ii) their non-zero elements are close to the main diagonal. More precisely, the following definition of a banded matrix can be used.

Assume that the matrix is such that its non-zero elements are located close to the main diagonal so that

$$(7.2) \quad a_{ij} = 0 \quad \text{if} \quad i-j > \text{NBANDL} \quad \text{or} \quad j-i > \text{NBANDU},$$

where $i=1,2,\dots,n$, $j=1,2,\dots,n$. Such a matrix is called banded if NBANDL and NBANDU are two integers which are, both of them, much smaller than n , where n is the order of the matrix. The parameter NBANDL gives the number of diagonals in the matrix band that are under the main diagonal, while NBANDU is the corresponding number of diagonals that are over the main diagonal. All matrix elements that are outside the matrix band are equal to zero. The matrix elements that are inside the matrix band may be non-zero elements, which means that not all of them are necessarily zeros. In fact, in many cases many of them are also zeros; see, for example, Fig. 7.3.

Assume now that systems of linear algebraic equations with banded coefficient matrices are to be solved. If the systems of linear algebraic equations are to be solved directly, then some efficient implementation of the well-known Gaussian elimination is to be applied. If the systems of linear algebraic equations are to be solved iteratively, then several basic linear algebra operations (such as matrix-vector multiplications, inner products and calculations of different norms) are to be performed. In both cases, the direct solution and the iterative solution, one must remember that the matrices involved in the computations are banded and one must exploit this information in order to obtain an efficient computational process.

The situation does not change very much when the diffusion sub-model is considered. Fast Fourier Transforms (FFT's) can again be applied when some pseudospectral discretization is applied. If finite elements or finite differences are used in the space discretization, the situation is as above. However, it must be emphasized that the matrices that are involved in the computations are normally not only banded, but also symmetric and positive definite. This fact must also be exploited in order to increase the efficiency of the computational process.

The treatment of the third sub-model, the deposition sub-model, is nearly trivial for two reasons: (i) some simple assumptions concerning the deposition processes were made, and (ii) the solution is easy from a numerical point of view. The deposition sub-model is a system of ODE's in which the separate equations are independent. Moreover, these are linear ODE's and therefore could be solved analytically. The process is easily vectorizable (the vectorization of this sub-model can be achieved in an obvious way). It is not necessary to discuss in detail the vectorization of the deposition under the assumptions made here. On the other hand, it must be stressed that in many cases it will be necessary to describe the deposition processes by applying more complicated physics. This latter situation will not be discussed in detail either, because if more complicated physics are applied (especially for the wet deposition), then the deposition sub-model will become rather similar to the chemical sub-model which is discussed below.

The chemical sub-model can cause problems when a numerical method is to be selected. This sub-model leads to a system of ODE's which is normally stiff (or even very stiff). Special time-integration methods for stiff systems of ODE's are to be used, and it is not easy to find an efficient method. As a matter of fact, the question of finding good numerical methods for the chemical sub-model is still open. However, if a numerical method is selected, then it is rather easy, in principle at least, to vectorize it, because the chemical reactions involve only the species at a given grid-point (there are no interconnections between species from different grid-points). Therefore long loops can be organized along the grid-points. This

may be a difficult task, but once accomplished, these loops will vectorize on every vector computer.

This short discussion of the sub-models shows clearly that the most important tasks that have to be vectorized are

- (i) basic linear algebra operations (first and foremost operations involving computations with banded matrices)

and

- (ii) fast Fourier transforms.

Of course, the other operations mentioned above must also be vectorized, but this is a problem that can easily be solved (in the most cases the compiler will automatically vectorize the other operations).

Sometimes the computations in the vertical direction are separated from the computations on the horizontal planes (see Section 3.1). This case will not be discussed in this chapter, because the situation does not change too much. In other words, in most cases the separation of the computations in the vertical direction from the computations on the horizontal planes does not lead to basic tasks that are different from the tasks that appear in the treatment of the previous sub-models.

It is necessary to explain why it is very important to vectorize a large air pollution code. The easiest way is to illustrate the ideas by analyzing the situation in a simple example. Let us assume that the AMDAHL VP1100 computer is to be used in the computations. Consider also the matrix-vector multiplication, where the matrix is assumed to be banded. This operation is very important when systems of linear algebraic operations are solved by iterative methods. If this operation is not vectorized (i.e. if the computer runs it in a scalar mode), then the speed of computations is approximately 8 MFLOPS¹. It will be shown in this chapter that if this arithmetic operation is properly vectorized, then the speed of computations is increased considerably; up to 240 MFLOPS. This means that the speed of computations can be accelerated by a factor up to 30 when the code is vectorized correctly. The speed-up will, of course, not always be so high, but nevertheless the vectorization of a code will normally result in a considerable acceleration of the speed of computations. The acceleration of the speed of computations is one of the crucial tasks when large air pollution problems are to be treated on computers (because the discretization of these models leads to the solution of very large numerical problems; see Section 5.1).

¹ **MFLOPS**: million floating point operations (additions and/or multiplications) per second

7.2. Using vectorized basic linear algebra operations

Both solving systems of linear algebraic equations, LAE's, with banded matrices and performing some other basic linear algebra operations with banded matrices are time-consuming and must be carried out in an efficient way when large LRTAP models are to be treated numerically on big modern vector processors (see the previous section). Such operations appear in many other fields of science and engineering (although the matrices involved are not necessarily banded). Therefore, it is necessary to develop kernels that are standard and, moreover, perform these operations efficiently. Such kernels are called basic linear algebra subprograms, BLAS's. BLAS's have been proposed first in Lawson et al. [284], and after that an extended set of BLAS's have been proposed in Dongarra et al. [113].

Let us start by considering computations that involve dense matrices. A matrix is called dense if either all its elements are non-zeros or there are zero elements but it is not efficient to exploit this fact and the zero elements are treated in the same way as the non-zeros during computations. Subroutines for dense matrices, which are vectorized and by which a high level of performance can be achieved on many vector machines, can be found in Dongarra et al. [113]. These subroutines are based on principles described, for example, in Daly and Du Croz [96], Du Croz and Wasniewski [122], Dongarra and Eisenstat [114], Dongarra et al. [115]. The problems connected with the development of kernels involving dense matrices are well-understood now. BLAS's for dense matrices exist at many computing centres. Many of them are written in machine language and it is both easy and efficient to attach them to big application packages.

The situation changes when operations with banded matrices are considered instead of operations with dense matrices. In order to illustrate this, let us consider the calculation of $y := y + Ax$, where A is a real matrix of order n , while x and y are vectors with n real components. This is a typical algebraic operation that occurs in the numerical treatment of many large scientific and engineering problems. Assume that the speed of computations is measured in MFLOPS. Assume also that the computations are carried out on a CRAY X-MP48 (by using one processor only) and that $n=1024$. Then the subroutine from the set of subprograms in Dongarra et al. [113] (this set is commonly called BLAS 2) performs the above operation with a speed greater than 160 MFLOPS when the matrix is dense, while a speed of only about 30 MFLOPS can be obtained by the corresponding subroutine for symmetric, positive definite and banded matrices when a matrix of the same order, $n=1024$, and a bandwidth of 32 is treated. It is clear that if the problem solved is very large, then it is highly desirable to be able both to exploit the special properties of matrix A (which leads to a reduction of the number of simple arithmetic operations) and to keep the speed of computations at the same level as the speed of computations for dense matrices (which results in a better exploitation of the power of the vector processors). This discussion together with the fact that banded matrices do appear very often in large-scale scientific computations explains why fast subroutines are needed for performing basic linear algebra operations for banded matrices. Examples of such subroutines will be described in this chapter.

A careful study of the matrices that appear after discretizing PDE's shows that these matrices are not only banded, but very often they also have many diagonals, within the matrix band, that contain no non-zero element (all their components are zeros). Thus, the matrix bands are very often sparse (which means, roughly speaking, that they contain a lot of zero elements) and, moreover, non-zero elements do not, in general, occur on random places (these fill whole, or at least nearly whole, diagonals). This is a special kind of sparsity, which can be exploited very efficiently and some algorithms, by which the sparsity of the matrix bands is exploited, will also be described in this chapter (together with the subroutines where these algorithms are implemented).

7.3. Vectorizing matrix-vector multipliers for banded matrices

Consider the following basic linear algebraic operation (multiply a matrix by a vector and add the result to another vector):

$$(7.3) \quad y := y + Ax,$$

where both x and y are vectors with n components and A is a square matrix of order n . If matrix A is a dense matrix, then the most straightforward way to perform the operation given by (7.3) is to multiply (for $i=1,2,\dots,n$) the elements in the i 'th row of matrix A with the corresponding components of vector x and to add this sum to the i 'th component of vector y . Thus, the new value of y_i is obtained by

$$(7.4) \quad y_i := y_i + \sum_{j=1}^n a_{ij} x_j.$$

This approach works very well also when matrix A is banded and when the calculations are carried out on a sequential computer. In the latter case the summation is carried out from $\max(1,j-\text{NBANDL})$ to $\min(j+\text{NBANDU},n)$.

In the algorithms for performing the matrix-vector multiplication that are based on (7.4), the calculations are organized by rows. The calculations can also be organized by columns. This is done, not only for operation (7.4) but for many other operations involving matrices, in the subroutines of LINPACK (see Dongarra et al. [112]). The algorithms in which the calculations are organized by columns perform (on many computers) better than the corresponding algorithms in which the calculations are organized by rows, because contiguous arrays are used in the former algorithms.

However, if the matrix is banded and if a vector computer is in use, then the speed of computations, measured in MFLOPS, achieved by either of the above two approaches, is

considerably slower than the speed of computations achieved by the corresponding algorithms for dense matrices. This can easily be established by running the appropriate subroutines from the BLAS 2 set (see Dongarra et al. [113]). It should be emphasized here that the computing times for the banded subroutines will normally be smaller than the computing times for the dense subroutines (although the latter are faster). This is so because the number of arithmetic computations is reduced very considerably when the bandedness is exploited. Nevertheless, it is desirable to try to achieve the same speed of computations for the algorithms that deal with banded matrices as that for the algorithms by which dense matrices are treated. Several efficient algorithms for matrix-vector multiplication in the case where the matrix is banded will be described below.

7.3.1. Vectorizable matrix-vector multipliers

The conventionally used algorithms for performing matrix-vector multiplications do not work very well on vector computers because the arrays involved in the loops are short when the matrix band is narrow (when $\text{NBANDL} + \text{NBANDU} + 1$ is a small integer) even in the case where the matrix itself is very large. In order to get long arrays in the loops, also when the matrix band is very narrow, one should organize the calculations by diagonals. Not only is the computing time reduced very considerably when this is done properly, but also the speed of computations becomes comparable with the speed of computations for the corresponding dense algorithms. The idea is very simple. Its realization in a computer code is demonstrated in Fig. 7.1, where the main loops used are given. The corresponding loop in the BLAS 2 subroutine for performing (7.3) is given in Fig. 7.2.

```

      DO 10 I=1,N
        Y(I)=Y(I)+A(I,NBANDL+1)*X(I)
      10 CONTINUE
C
      DO 30 J=1,NBANDU
        IEND=N-J
        JJ=NBANDU+1+J
        DO 20 I=1,IEND
          Y(I)=Y(I)+A(I,JJ)*X(I+J)
        20 CONTINUE
      30 CONTINUE
C
      DO 50 J=1,NBANDL
        ISTART=NBANDL+2-J
        DO 40 I=ISTART,N
          Y(I)=Y(I)+A(I,J)*X(I+1-ISTART)
        40 CONTINUE
      50 CONTINUE

```

Figure 7.1
The main loops of matrix-vector multiplier
in which the computations are organized by diagonals.

The two codes show clearly that the arrays involved in the inner loops are long in the first code, whereas they will normally be rather short in the second code. Let $\text{NBANDL} = \text{NBANDU} = 32$ and $N = 1024$. Then the length of the arrays in the first code is about 1000, whereas the arrays involved in the inner loop of the second code are at most of length 65.

It is necessary to explain how the banded matrices are to be stored when the two codes, given in Fig. 7.1 and Fig. 7.2, are used. This can be illustrated by applying a small matrix selected from the set of the test-matrices that will be used in the numerical experiments. Therefore a class of test-matrices will first be introduced in the next sub-section and then a small matrix within this class will be used to present the storage structures appropriate for the calculations sketched in the codes from Fig. 7.1 and Fig. 7.2.

```

NBAND1=NBANDU+1
DO 20 J=1,N
  TEMP=X(J)
  NBAND2=NBAND1-J
  DO 10 I=MAX0(1,J-NBANDU),MIN0(N,J+NBANDL)
    Y(I)=Y(I)+TEMP*A(NBAND2+I,J)
  10 CONTINUE
20 CONTINUE

```

Figure 7.2
The main loops in the BLAS 2 matrix-vector multiplier
for general banded matrices.

It is necessary to mention here that calculations by diagonals have also been considered, in another context, by Madsen et al. [303].

7.3.2. A class of banded test-matrices

It is desirable to have a class of banded test-matrices for which the following properties are satisfied:

- (1) matrices of different orders can be generated,
- (2) all matrices from the class are generated by a simple and short program,
- (3) the matrices are very similar to matrices that often appear in practice.

If the first of these requirements is satisfied, then it is possible to study systematically the variations in the performance of different codes when the order of the matrices is varied. The

second requirement will allow the scientists to start the experiments quickly without using a lot of time to attach complicated programs to the tested code. If the third requirement holds, then there is a good reason to expect that the code tested will also perform well in the solution of scientific and engineering problems from different fields.

A class of test-matrices, for which the three requirements (1)-(3) are satisfied, can be constructed as follows. Let us assume that the bandwidth NBANDL of the desired matrix is already chosen. Set $\text{NBANDU} = \text{NBANDL}$ and $N = (\text{NBANDL})^2$. Let d be an arbitrary real number. Consider a matrix of order N such that

- (a) All its elements on the main diagonal are equal to four.
- (b) The elements in the diagonals whose first elements are a_{12} and a_{1k} with $k = \text{NBANDU} = \text{NBANDL}$ are equal to $-1-d$.
- (c) The elements in the diagonals whose first elements are a_{21} and a_{k1} with $k = \text{NBANDL}$ are equal to $-1+d$.
- (d) All other elements are equal to zero.

In this way a class of test-matrices depending on two parameters is constructed. If NBANDL is varied, then matrices of different orders are constructed. If $d = 0.0$, then matrices similar to well-known matrices (that appear in the discretization of the Laplace operator by the five-point rule) are obtained. These matrices are symmetric and positive definite. If $d \neq 0.0$, then the matrices are not symmetric and the positive definiteness is in general also lost. In the latter case, the matrices are similar to matrices that appear in the discretization of PDE's containing both first-order and second-order space derivatives.

A simple program that produces matrices of this class has been developed. This program will be used in the numerical experiments in this chapter (but it should be mentioned here that the conclusions were drawn by using also some results that were obtained by running other matrices on various vector computers). For any pair of parameters (NBANDL, d) the program produces a matrix according to the rules (a)-(d). NBANDL must be a positive integer that is greater than 1.

Now it is easy to show how the bandedness is exploited to store the matrices, so that the codes given in the previous sections can be used. Let $\text{NBANDL}=4$ and $d=4$. The matrix in the family of test-matrices corresponding to these two parameters is given in Fig. 7.3.

If the bandedness of matrix A is not exploited, then A is normally stored in a two-dimensional array, where the rows of the matrix are rows in the array and the columns of the matrix are columns in the array. The structure should be packed when the bandedness is to be exploited. Using the well-known structure adopted in the LINPACK subroutines (Dongarra et al. [112]), the matrix from Fig. 7.3 can be stored as in Fig. 7.4.

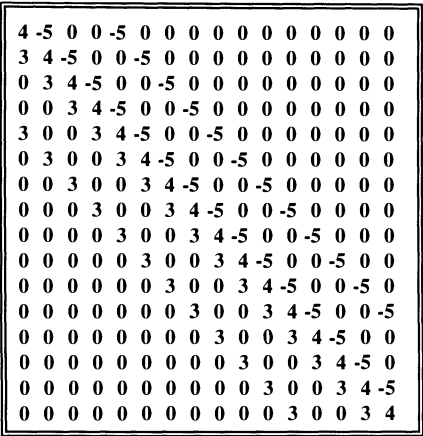


Figure 7.3
A test-matrix with $n=16$, $NBANDL=4$ and $NBANDU=4$.

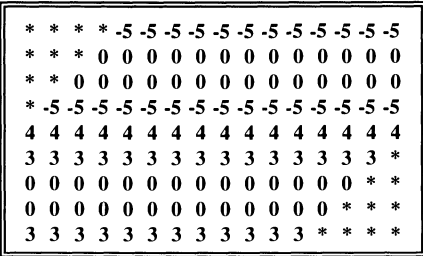


Figure 7.4
The matrix from Fig. 7.3 stored by using the LINPACK scheme.

The matrix structure in Fig. 7.4 is used in the banded matrix multipliers in BLAS 2 (see Dongarra et al. [113]). In these subroutines the calculations are organized by using the columns of matrix A and, more precisely, the parts of the columns of A that are within the matrix band. This means that if the matrix is with a narrow band, then one is working with short arrays. Therefore it is clear that it is better to reorganize the calculations so that the computations in inner loops have to be carried out by diagonals. In order to get contiguous vectors, it is also useful to store the diagonals as columns in the array in which the matrix band of A is packed. This leads to the structure shown in Fig. 7.5.

The zero diagonals are stored in the schemes given in Fig. 7.4 and Fig. 7.5. Of course, this is not necessary. Both storage and computing time can be saved by storing the non-zero diagonals only. The storage scheme in which only the non-zero diagonals are stored is illustrated in Fig. 7.6.

If the diagonals of the banded matrix are packed as in Fig. 7.6, then it is necessary to use two extra arrays in which the distances of the diagonals to the main diagonal are given. The lengths of these arrays are equal to the numbers of diagonals in the lower matrix band and the upper matrix band respectively (i.e. to NBANDL and NBANDU). The contents of these arrays are given in Fig. 7.6 over the corresponding diagonals. For the particular matrices considered here the information about the distances of the diagonals to the main diagonal is quite symmetric, but it is clear that in general this is not true.

For symmetric matrices one has to store only the upper (or the lower) band, and therefore an extra array will be sufficient to provide the information that is needed in order to locate the positions of the diagonal in relation to the main diagonal.

The packed storage scheme illustrated in Fig. 7.6 can only be used together with some indirect addresses. The use of such addresses normally leads to a great decrease in the speed of computations when this occurs in inner loops. By a careful organization of the computations one can avoid using indirect addresses in the inner loops. Thus, indirect addressing is only used in the outer loops which do not vectorize anyway.

Now everything is prepared for the experiments, in which the usefulness of the reorganization of the computations (so that these are carried out by diagonals) will be demonstrated. The numerical experiments are described in the following sub-sections.

*	*	*	*	4	-5	0	0	-5
*	*	*	3	4	-5	0	0	-5
*	*	0	3	4	-5	0	0	-5
*	0	0	3	4	-5	0	0	-5
3	0	0	3	4	-5	0	0	-5
3	0	0	3	4	-5	0	0	-5
3	0	0	3	4	-5	0	0	-5
3	0	0	3	4	-5	0	0	-5
3	0	0	3	4	-5	0	0	-5
3	0	0	3	4	-5	0	0	-5
3	0	0	3	4	-5	0	0	-5
3	0	0	3	4	-5	0	0	-5
3	0	0	3	4	-5	0	0	*
3	0	0	3	4	-5	0	*	*
3	0	0	3	4	-5	*	*	*
3	0	0	3	4	*	*	*	*

Figure 7.5.
The matrix from Fig. 7.3 stored by diagonals.

4	1	-	1	4
*	*	4	-5	-5
*	3	4	-5	-5
*	3	4	-5	-5
*	3	4	-5	-5
3	3	4	-5	-5
3	3	4	-5	-5
3	3	4	-5	-5
3	3	4	-5	-5
3	3	4	-5	-5
3	3	4	-5	-5
3	3	4	-5	-5
3	3	4	-5	*
3	3	4	-5	*
3	3	4	-5	*
3	3	4	*	*

Figure 7.6.
The matrix from Fig. 7.3 when only the non-zero diagonals are stored.
The distances of the non-zero diagonals to the main diagonal are given
at the top of the figure.

7.3.3. Performance of matrix-vector multiplications

The performance of the two matrix-vector multiplications, the main loops of which are given on Fig.7.1 and Fig. 7.2, will be studied in this section. The codes will be called DIAG 0 and BLAS 2 respectively. It should be mentioned here that BLAS 2 is in fact a simplified version of the subroutine SGBMV from Dongarra et al. [113] which performs directly (7.3) instead of the operation $y:=ay+bAx$ (this means that the fact that $a = b = 1.0$ has been exploited in order to improve the performance of SGBMV).

The matrices from the class banded test-matrices that has been described in the previous sub-section were run with NBANDL=8,16,24,...,96 and with d=4 on an AMDAHL VP1100. The speeds of the computations, measured in MFLOPS, for the two codes are given in Table 7.1. The bandwidth parameter NBAND = NBANDL + NBANDU + 1, which is used in Table 7.1, is the actual length of the arrays in the inner loops of BLAS 2. The computing times are proportional to the numbers of MFLOPS (because the same number of arithmetic operations is needed to calculate the matrix-vector product in both algorithms) and, therefore, there is no need to list them.

The results shown in third and the fourth columns of Table 7.1 indicate that the calculations by diagonals are to be preferred even when the matrices are very large (and, in our examples, also the matrix band becomes rather large if the matrix is large). The results can be further improved when a special device, unrolling, is applied. Unrolling will be studied in the next section.

N	NBAND	BLAS 2	DEPTH 0	DEPTH 2	DEPTH 4	DEPTH 8	DEPTH 16
64	17	11	44	36	31	30	30
256	33	22	80	120	99	103	101
512	49	33	85	128	135	148	147
1024	65	44	91	137	166	185	195
1600	81	55	110	132	172	195	204
2304	97	68	114	139	182	208	224
3136	113	73	116	137	184	213	226
4096	129	76	115	140	189	219	237
5184	145	79	119	141	192	222	238
6400	161	83	121	140	192	222	241
7744	177	86	121	140	194	225	242
9216	193	89	119	141	194	226	245

Table 7.1

Performing matrix-vector multiplications with different algorithms.

The speed of computations is measured in MFLOPS on AMDAHL VP1100.

The matrices used are of the same type as the matrix given in

Fig. 7.3 with $NBANDL = NBANDU = 4k$ ($k=2,4,6,\dots,24$), $N = (NBANDL)^2$

and $NBAND = NBANDL + NBANDU + 1$. The sparsity of the matrix band is

not exploited in these runs. DEPTH 0 means that no unrolling is

used. Unrolling is not applied in the BLAS 2 kernel-subroutine.

7.3.4. Unrolling double loops

On some vector computers a special device, called "unrolling" (see, for example, Dongarra et al. [115] or Dongarra and Eisenstat [114]) can successfully be applied to increase the speed of the computations. This device is efficient for double loops and, roughly speaking, leads to performing a greater amount of computational work per iteration of the inner loop. Unrolling can be carried out when the arrays involved in the inner loop are equally long. A close examination of the inner loops in the code given in Fig. 7.1 shows that this requirement is not satisfied when the computations are carried out by diagonals. Therefore, it is necessary to perform some preliminary work in order to prepare the code in Fig. 7.1 for unrolling. The idea is to perform separately the computations in the upper triangle of the lower band (for the example shown in Fig. 7.5 the elements in rows 2, 3 and 4 which are located to the left of the main diagonal). This part of the computational work is carried out, in a separate loop, without unrolling. Then unrolling is carried out for the remaining part of the lower band. The arrays in this remaining part are equally long. Moreover, if the matrix under consideration is large, then the elements of the former part of

the lower band are a few only, while many more elements are located in the part of the lower band that can be unrolled.

Similar ideas are to be applied in connection with the upper band (in this case the small loop, which cannot be unrolled, contains elements in the bottom of Fig. 7.3 that are to the right of the main diagonals). The implementation of these ideas is illustrated in the code given in Fig. 7.7.

The two major loops in the code given in Fig. 7.7 can easily be unrolled. In the case where the depth of unrolling is two, this is done in the code given in Fig. 7.8. It is obvious how unrolling with higher depths can be obtained by applying the same ideas as those used to obtain the code in Fig. 7.8.

```

      DO 10 I=1,N
        Y(I)=Y(I)+A(I,NBANDL+1)*X(I)
      10 CONTINUE
C
      ISTART=N-NBANDU+1
      DO 30 J=1,NBANDU-1
        IEND=N-J
        JJ=NBANDU+1+J
        DO 20 I=ISTART,IEND
          Y(I)=Y(I)+A(I,JJ)*X(I+J)
        20 CONTINUE
      30 CONTINUE
C
      DO 50 J=1,NBANDU
        JJ=NBANDU+1+J
        DO 40 I=1,N-NBANDU
          Y(I)=Y(I)+A(I,JJ)*X(I+J)
        40 CONTINUE
      50 CONTINUE
C
      DO 70 J=2,NBANDL
        ISTART=NBANDL+2-J
        DO 60 I=ISTART,NBANDL
          Y(I)=Y(I)+A(I,J)*X(I+1-ISTART)
        60 CONTINUE
      70 CONTINUE
C
      DO 90 J=1,NBANDL
        JJ=NBANDL+1-J
        DO 80 I=NBANDL+1,N
          Y(I)=Y(I)+A(I,J)*X(I-JJ)
        80 CONTINUE
      90 CONTINUE

```

Figure 7.7

The main loops of matrix-vector multiplier
in which the computations are prepared for unrolling.

Experiments, in which unrolling with several depths has been applied, have been carried out both on a CRAY X-MP48 computer and on an AMDAHL VP1100 computer. Some results obtained with unrolling with depths 2, 4, 8 and 16 are given in Table 7.1. It is clear that for sufficiently large values of N the unrolling with depth 16 performs about twice as fast as the case where no unrolling is carried out. It should be pointed out that unrolling with depth 32 is slower than unrolling with depth 16. In the experiments on the CRAY X-MP machine unrolling with depth 8 gave best performance.

7.3.5. Exploiting the sparsity of the matrix band

The sparsity of the matrix band is not exploited in the experiment, the results of which are shown in Table 7.1. Of course, this is not the best choice. Much better computing times can be achieved by exploiting the sparsity of the matrix band (using the structure shown on Fig. 7.7), although the speed of the computational process, measured in MFLOPS, is decreased when the sparsity is exploited. This is demonstrated in Table 7.2 and in Table 7.3. Unrolling the outer loops has no great effect for the particular matrices that are used here, because there are only two non-zero diagonals on each side of the main diagonal. However, if the number of non-zero diagonals is larger, then the efficiency of unrolling, measured in MFLOPS, will become greater also when the sparsity of the matrix band is exploited.

Compare the results for DEPTH 0 (no unrolling) in Table 7.2 with the corresponding results in Table 7.1. The comparison shows that the speed of computations is decreased by approximately 10% - 20% when the sparsity is exploited. This is caused by the use of indirect indices in the sparse code. Fortunately, indirect indices are to be used only in the outer loops (which are not vectorized anyway). This is the reason for the very moderate decrease of the speed of computations with the sparse code.

There are only two non-zero diagonals both in the upper band and in the lower band. Therefore unrolling with depth greater than 2 should practically have no effect on the speed of computations and the speeds of computations obtained with all unrolling in the sparse code should be compared with the speed of computations obtained with DEPTH 2 (unrolling with depth 2) in Table 7.1. Taking this into account, it is seen that the relationship between the speeds of computations for the dense code and the sparse code when unrolling is carried out is the same as the relationship between the speeds of computations for the dense code and the sparse code when unrolling is not performed (see the previous sub-section).

```

    ISTART=N-NBANDU+1
    DO 30 J=1,NBANDU-1
        IEND=N-J
        JJ=NBANDU+1+J
        DO 20 I=ISTART,IEND
            Y(I)=Y(I)+A(I,JJ)*X(I+J)
20      CONTINUE
30      CONTINUE
C
    J=MOD(NBANDU,2)
    IF(J.GE.1) THEN
        JJ=NBANDU+1+J
        DO 40 I=1,N-NBANDU
            Y(I)=Y(I)+A(I,JJ)*X(I+J)
40      CONTINUE
        END IF
C
    JSTART=J+2
    DO 60 J=1,NBANDU,2
        JJ=NBANDU+1+J
        DO 50 I=1,N-NBANDU
            Y(I)=Y(I)+A(I,JJ)*X(I+J)+A(I,JJ-1)*X(I+J-1)
50      CONTINUE
60      CONTINUE
C
    DO 80 J=2,NBANDL
        ISTART=NBANDL+2-J
        DO 70 I=ISTART,NBANDL
            Y(I)=Y(I)+A(I,J)*X(I+1-ISTART)
70      CONTINUE
80      CONTINUE
C
    J=MOD(NBANDL,2)
    IF(J.GE.1) THEN
        JJ=NBANDL+1-J
        DO 90 I=NBANDL+1,N
            Y(I)=Y(I)+A(I,J)*X(I-JJ)
90      CONTINUE
        END IF
C
    DO 110 J=1,NBANDL
        JJ=NBANDL+1-J
        DO 100 I=NBANDL+1,N
            Y(I)=Y(I)+A(I,J)*X(I-JJ)+A(I,J-1)*X(I-JJ-1)
100     CONTINUE
110     CONTINUE

```

Figure 7.8

Unrolling with depth 2 for the code in Fig. 7.7. The contribution due to the main diagonal, which is not given here, can be calculated as in the code on Fig. 7.7 (see loop 10 on Fig. 7.7).

N	NBAND	BLAS 2	DEPTH 0	DEPTH 2	DEPTH 4	DEPTH 8	DEPTH 16
64	17	11	25	19	19	18	17
256	33	22	57	55	55	53	51
512	49	33	71	82	83	80	79
1024	65	44	84	99	104	102	100
1600	81	55	95	107	113	112	110
2304	97	68	99	118	123	120	120
3136	113	73	103	121	127	125	125
4096	129	76	106	125	132	134	133
5184	145	79	108	126	135	133	133
6400	161	83	109	128	136	135	135
7744	177	86	110	128	138	138	136
9216	193	89	110	129	139	139	138

Table 7.2

Performing matrix-vector multiplications with different algorithms.

The speed of computations is measured in MFLOPS on AMDAHL VP1100.

The matrices used are of the same type as the matrix given in Fig. 3.1 with $NBANDL = NBANDU = 4k$ ($k=2,4,6,\dots,24$), $N = (NBANDL)^2$ and $NBAND = NBANDL + NBANDU + 1$. The sparsity of the matrix band is exploited in these runs. DEPTH 0 means that no unrolling is used. Unrolling is not applied in the BLAS 2 kernel-subroutine.

The comparison of the speeds of computations of the sparse code and the dense code is not fair with regard to the sparse code. This is so because the number of arithmetic operations is dramatically reduced when the sparse code is in use. Therefore, it is also useful to show the computing times obtained in the runs, which is done in Table 7.3.

7.3.6. Multitasking on multiprocessors

It has been mentioned in Chapter 6 that many vector computers have several processors that can work in parallel. If this is the case, then it is important to exploit the possibility of running some parts of the code concurrently. This possibility will be discussed briefly in this section. The parallel computations will be discussed in more detail in Chapter 8 and Chapter 9).

The codes, which have been developed in the previous sections, have a very nice property: it is nearly obvious how to modify them in order to achieve parallel computations

on a multiprocessor. Let us assume that the number of the processors available is NCPU. Assume also that two integer arrays, IBGN and ISTOP are available and that both are of length NCPU. If the banded matrix that is to be treated is of order N, then the contents of these two arrays are to be initialized as follows:

(7.5)
$$IBGN(I) = [(I-1) * N / NCPU] + 1, \quad ISTOP(I) = [I * N / NCPU],$$
$$I = 1, 2, \dots, NCPU.$$

N	NBAND	BLAS 2	Sparsity not exploited	Sparsity exploited
64	17	0.20	0.07 (35.0%)	0.03 (15.0%)
256	33	0.76	0.17 (22.3%)	0.05 (6.6%)
512	49	1.71	0.38 (22.2%)	0.07 (4.1%)
1024	65	3.03	0.68 (22.4%)	0.10 (3.3%)
1600	81	4.73	1.26 (26.6%)	0.14 (3.0%)
2304	97	6.83	1.99 (29.1%)	0.19 (2.8%)
3136	113	9.80	3.13 (31.9%)	0.24 (2.4%)
4096	129	14.01	4.43 (31.6%)	0.31 (2.2%)
5184	145	19.05	6.27 (32.9%)	0.39 (2.0%)
6400	161	24.90	8.55 (34.3%)	0.47 (1.9%)
7744	177	31.99	11.34 (35.4%)	0.56 (1.8%)
9216	193	40.17	14.50 (36.1%)	0.66 (1.6%)

Table 7.3
Performing matrix-vector multiplications for banded matrices.
The computing times, measured in seconds on AMDAHL VP1100.
The results obtained when unrolling with depth 16 is applied
are given in the case where the sparsity is not exploited.
When the sparsity is exploited, the results given in this table
are obtained using unrolling with depth 4. In all cases 1000
matrix-vector products are calculated.

Assume now that a CRAY X-MP computer is to be used. Consider the long loops in Fig. 7.7; i.e. consider Loop 10, Loop 50 and Loop 90. The main part of the computational work is carried out in these loops. This work can be performed in parallel by using the code in Fig. 7.9. The other codes can also be prepared for parallel computations using the same ideas. Moreover, the same ideas can be used not only in connection with the CRAY X-MP computer but also on some other architectures. Some codes for symmetric banded matrices are discussed in Zlatev et al. [547]. In Zlatev et al. [547] it is shown the speed up parameters of the code based on the ideas sketched above are rather good (being in the interval [3.5,3.85] for sufficiently large matrices when NCPU=4).


```

CMIG$ DOGLOBAL
DO 60 K=1,NCPU
  K1=IBGN(K)
  K2=ISTOP(K)
  KK1=MIN(K1,NBANDL+1)
  KK2=MAX(K2,N-NBANDU)
  DO 10 I=K1,K2
    Y(I)=Y(I)+A(I,NBANDL+1)*X(I)
  10 CONTINUE
  DO 30 J=1,NBANDU
    JJ=NBANDU+1+J
    DO 20 I=K1,KK2
      Y(I)=Y(I)+A(I,JJ)*X(I+J)
    20 CONTINUE
  30 CONTINUE
  DO 50 J=1,NBANDL
    JJ=NBANDL+1-J
    DO 40 I=KK1,K2
      Y(I)=Y(I)+A(I,J)*X(I-JJ)
    40 CONTINUE
  50 CONTINUE
  60 CONTINUE

```

Figure 7.9

One of the main loops of matrix-vector multiplier,
in which the computations are organized by diagonals,
prepared for parallel execution on a CRAY X-MP computer.

7.4. Vectorizing the solution of systems with banded matrices

One should use special methods when systems of linear algebraic equations (LAE's) whose matrices are banded are to be solved, because this normally leads to a considerable decrease of the amount of the computational work. Furthermore, if the matrices involved in the calculations are symmetric and positive definite, then this fact should also be exploited during the solution process. Finally, it is not very easy to decide what kind of method (a direct method or an iterative method) will perform best.

7.4.1. Solving symmetric and positive definite systems of linear algebraic equations by conjugate gradient methods

Systems of linear algebraic equations (LAE's) whose coefficient matrices are band, symmetric and positive definite can efficiently be solved by using a conjugate gradients algorithm with some simple preconditionings. The advantages of using this iterative procedure

are described in Zlatev et al. [547]. Numerical results from some experiments, which have been performed on an AMDAHL VP1100, will be presented in this section. The matrices used in the experiments are obtained from the family of band test-matrices introduced in § 7.3.2 by setting the parameter d equal to zero. Thus, the test-matrices used are very similar to matrices that appear in the discretization of the Laplace operator on a square space domain by the five-point rule (which has already been mentioned in § 7.3.2).

The results obtained by the use of the conjugate gradients algorithm will be compared with results obtained by a direct method in which the bandedness, the symmetry and the positive definiteness are exploited. The purpose is to show that on vector machines an iterative method could perform better than a direct method even in the case where on a sequential computer the direct method performs better than the iterative one. This is so because the computations with banded matrices can be carried out very efficiently on a vector machine, while one is forced to work with short arrays during the factorization stage of the direct methods for banded matrices and, thus, the process cannot be vectorized on a high level of performance. The LINPACK subroutines SPOFA and SPOSL, see Dongarra et al. [112] have been used. It should be mentioned that a new package, LAPACK, has been developed by scientists from the Argonne Laboratory, the Courant Institute, the NAG Library and some other sites (Anderson et al. [8]). This means that some more efficient subroutines are now available. However, this will not alter too much the conclusions drawn in this section, because the reason for the poor performance of the direct methods for banded matrices on vector machines is due, first and foremost, to the work with rather short arrays during the factorization (and this does not depend on the particular implementation of the direct methods in subroutines). Nevertheless, one should have in mind the fact that better subroutines are available for solving LAE's by direct methods.

The speed of computations in the case where the sparsity of the matrix bands is not exploited is shown (for unrollings with different depths) in Table 7.4. The matrices from the family introduced in § 7.3.2 are in general non-symmetric, but (as mentioned in § 7.3.2) they become symmetric and positive definite matrices in the case where $d = 0.0$ is chosen. One may exploit the symmetry as well as the positive definiteness and this has been done by applying the algorithms described in Zlatev et al. [547].

It is seen, from Table 7.4, that the results are best when unrolling with depth 16 is applied. Moreover, the results are very similar, especially for large matrices, to the results presented in Table 7.1. Of course, this is not a great surprise: when the matrices are large and the sparsity of the matrix band is not exploited, then the matrix-vector multiplication is the dominating computational process during the solution of LAE's by iterative methods. This observation shows clearly why one should be very careful in the preparation of good kernel subroutines for matrix-vector multiplications.

Also in this case (as for matrix-vector multiplications; see the preceding sub-section) the sparsity of the matrix bands can very successfully be exploited. The speeds of computations obtained when the sparsity is exploited are presented in Table 7.5. The speed of computations is in general not reduced (in comparison with the speed of the computations for the case where the sparsity is not exploited) when no unrolling is carried out, because the matrix-vector multiplications do not dominate too much the computational process when the sparsity is exploited. The manipulation with one-dimensional arrays (calculations of vector

norms, copies and multiplications of vectors with constants) takes a considerable part of the computational process. These operations are performed with a high speed of computations, and thus the speed of computations for the global process is also increased.

N	NBAND	LINPACK	BLAS 2	DEPTH 0	DEPTH 2	DEPTH 4	DEPTH 8	DEPTH 16
64	17	1	7	36	29	30	29	28
256	33	2	12	79	87	94	101	100
512	49	2	17	84	112	135	145	145
1024	65	3	21	92	133	159	184	189
1600	81	4	24	104	140	172	188	198
2304	97	4	29	107	148	180	204	219
3136	113	5	32	111	147	183	208	221
4096	129	5	36	112	154	191	218	232
5184	145	6	38	112	154	189	218	233
6400	161	7	42	114	154	192	221	239
7744	177	7	45	114	156	193	221	239
9216	193	8	48	114	156	193	222	243

Table 7.4
The speed of computations, measured in MFLOPS on AMDAHL VP1100, obtained in the solution of systems of LAE's with symmetric and positive definite banded matrices by the use of a conjugate gradients algorithm. The sparsity of the matrix band is not exploited in these runs. DEPTH 0 means that no unrolling is used. Unrolling is not applied in the BLAS 2 kernel-subroutine.

Unrolling with depth greater than two has no great effect on the speed of computations. The reasons for this phenomenon are the same as those presented in connection with Table 7.2.

Again, as in the previous section, the comparison of the speeds of computations for the dense case (Table 7.4) and the sparse case (Table 7.5) is not very fair to the sparse case. Therefore it is illustrative to give also the computing times. This is done in Table 7.6. The results show very clearly the advantages of the exploitation of the sparsity in this situation.

The percents are calculated in connection with the computing times for the BLAS 2 subroutine. Note that the calculations by diagonals are even more efficient in this situation than in the case the matrices are not symmetric. This is so because the length of the arrays that are used in the inner loops in the code BLAS 2 is equal to the half bandwidth (the actual length of the arrays being $NBAND1 = (NBAND-1)/2$ where NBAND is given in the tables).

N	NBAND	LINPACK	BLAS 2	DEPTH 0	DEPTH 2	DEPTH 4	DEPTH 8	DEPTH 16
64	17	1	7	24	22	21	20	19
256	33	2	12	63	67	67	64	64
512	49	2	17	82	95	95	95	92
1024	65	3	21	103	121	121	119	118
1600	81	4	24	112	128	129	128	127
2304	97	4	29	119	138	140	140	139
3136	113	5	32	124	139	143	143	140
4096	129	5	36	126	149	146	147	146
5184	145	6	38	130	149	149	149	151
6400	161	7	42	131	151	153	153	152
7744	177	7	45	134	151	152	152	153
9216	193	8	48	134	155	155	154	152

Table 7.5
The speed of computations, measured in MFLOPS on AMDAHL VP1100, obtained in the solution of systems of LAE's by the use of a conjugate gradients algorithm. The sparsity of the matrix band is exploited in these runs. DEPTH 0 means that no unrolling is used. Unrolling is not applied in the BLAS 2 kernel-subroutine.

The results concerning the LINPACK subroutines show clearly that these subroutines perform in a scalar mode. Therefore the performance of these routines, compared with the vectorized iterative routines, is poorer. The results given in Table 7.4 - Table 7.5 are calculated with a requirement for achieving accuracy of order 10^{-3} . On AMDAHL VP1100 the computing times needed in the solution of the systems by iterative methods are smaller than those needed in the LINPACK subroutines even if an accuracy of order 10^{-10} is required. Accuracy of order 10^{-10} has been achieved (for the problem with $n=9216$) after 346 iterations. The computing time was 5.12 seconds when unrolling with depth 16 is applied (the computing time for the LINPACK subroutines being 11.47 seconds; see Table 7.5).

An error estimation is available when the iterative methods are used. The stopping criteria proposed in Zlatev and Nielsen [541] are applied. The performance of these criteria in the particular situation discussed here is illustrated in Table 7.7. The results shown in Table 7.7 indicate that the stopping criteria used are reliable and in many situations do ensure the required order of accuracy.

The LINPACK subroutines available on CRAY X-MP perform better (a vectorized version of LINPACK for CRAY computers has been prepared by the specialists at CRAY Research). Some loops are vectorized on this computer and therefore the performance of the LINPACK subroutines becomes in many situations better than the performance of the iterative subroutine based on the BLAS 2 kernel for calculating matrix-vector products. However, also on this computer the use of kernels in which the calculations are organized by

diagonals is to be preferred for the class of matrices used; the computing time being about twice smaller even if a high accuracy (10^{-10}) is required.

The situation changes when a scalar computer is in use. On vector computers a very high speed of computations can be achieved when the computations are organized by diagonals (because in this way one works with both long and contiguous arrays). On sequential computers this advantage is lost. The LINPACK subroutines work very efficiently on such computers and as a rule they will give best results for the problems from the family introduced in this chapter. This conclusion is based on experiments performed on a UNIVAC 1100 computer.

N	NBAND	LINPACK	BLAS 2	Sparsity not exploited	Sparsity exploited
64	17	0.01	0.01	< 0.01	< 0.01
256	33	0.05	0.06	0.01 (16.7%)	< 0.01
512	49	0.18	0.22	0.02 (9.1%)	0.01 (4.5%)
1024	65	0.42	0.52	0.06 (11.5%)	0.01 (1.9%)
1600	81	0.82	0.93	0.12 (12.9%)	0.02 (2.2%)
2304	97	1.43	1.63	0.21 (12.9%)	0.03 (1.8%)
3136	113	2.27	2.65	0.38 (14.3%)	0.05 (1.9%)
4096	129	3.39	3.77	0.58 (15.4%)	0.07 (1.9%)
5184	145	4.83	5.60	0.92 (16.4%)	0.10 (1.8%)
6400	161	6.63	7.65	1.36 (17.7%)	0.13 (1.7%)
7744	177	8.84	10.52	1.97 (18.7%)	0.18 (1.7%)
9216	193	11.47	13.46	2.68 (19.9%)	0.22 (1.6%)

Table 7.6

The computing times, measured in MFLOPS on AMDAHL VP1100, obtained in the solution of systems of LAE's with symmetric and positive definite banded matrices by the use of a conjugate gradients algorithm.

The results obtained when unrolling with depth 8 is applied are given in the case where the sparsity is not exploited.

When the sparsity is exploited, the results given in this table are obtained using unrolling with depth 4. The numbers given in are obtained by using the results for the BLAS 2 kernel.

The classical conjugate gradients algorithm has been used here as an iterative procedure (see Hestenes and Stiefel [236] or Reid [362]). Other iterative methods for symmetric and positive matrices, as those discussed in Evans [160-163], Rice and Boisvert [366], Varga [456], Vinsome [464], Washpress [467], Young [493-496] or Young and Kincaid [498,499], can also be applied. Several other iterative methods have been tested in connection with the air pollution models described in the previous chapters by Hov et al. [246].

Different preconditionings proposed in the above references can be attached to the iterative algorithms selected. In Zlatev et al. [547] a preconditioning proposed in Dubois et al. [121] has been used. An acceleration of the Jacobi method, proposed by Martins [310], has been attempted in Zlatev et al. [547].

N	NBAND	Estimated errors		Exact errors		
		BLAS 2	DIAGONALS	LINPACK	BLAS 2	DIAGONALS
64	17	4.6*10 ⁻⁴	4.6*10 ⁻⁴	3.5*10 ⁻¹⁴	6.4*10 ⁻⁵ (18)	6.4*10 ⁻⁵ (18)
256	33	7.8*10 ⁻⁴	7.8*10 ⁻⁴	3.3*10 ⁻¹³	5.0*10 ⁻⁵ (37)	5.0*10 ⁻⁵ (37)
512	49	2.2*10 ⁻⁴	2.2*10 ⁻⁴	1.0*10 ⁻¹²	5.3*10 ⁻⁵ (56)	5.3*10 ⁻⁵ (56)
1024	65	1.7*10 ⁻⁴	1.7*10 ⁻⁴	1.2*10 ⁻¹²	6.8*10 ⁻⁵ (74)	6.8*10 ⁻⁵ (74)
1600	81	8.6*10 ⁻⁴	8.6*10 ⁻⁴	4.8*10 ⁻¹²	8.3*10 ⁻⁴ (82)	8.3*10 ⁻⁴ (82)
2304	97	7.6*10 ⁻⁴	7.6*10 ⁻⁴	8.1*10 ⁻¹²	7.6*10 ⁻⁴ (98)	7.6*10 ⁻⁴ (98)
3136	113	9.6*10 ⁻⁴	9.6*10 ⁻⁴	1.4*10 ⁻¹¹	7.5*10 ⁻⁴ (113)	7.5*10 ⁻⁴ (113)
4096	129	9.5*10 ⁻⁴	9.4*10 ⁻⁴	2.0*10 ⁻¹¹	1.1*10 ⁻³ (121)	1.1*10 ⁻³ (121)
5184	145	9.3*10 ⁻⁴	9.3*10 ⁻⁴	2.8*10 ⁻¹¹	9.7*10 ⁻⁴ (137)	9.7*10 ⁻⁴ (137)
6400	161	9.4*10 ⁻⁴	9.4*10 ⁻⁴	4.0*10 ⁻¹¹	9.3*10 ⁻⁴ (151)	9.3*10 ⁻⁴ (151)
7744	177	9.1*10 ⁻⁴	9.1*10 ⁻⁴	5.3*10 ⁻¹¹	9.1*10 ⁻⁴ (166)	9.1*10 ⁻⁴ (166)
9216	193	9.3*10 ⁻⁴	9.3*10 ⁻⁴	7.7*10 ⁻¹¹	1.0*10 ⁻³ (177)	1.0*10 ⁻³ (177)

Table 7.7
The accuracy achieved in the solution of LAE's with symmetric and positive definite matrices. DIAGONALS refer to any of the algorithms used in the previous tables (unrolling having no influence on the accuracy achieved). The numbers of iterations needed to achieve the accuracy required are given in brackets. It is seen that the routines based on the BLAS 2 kernels and computations by diagonals give identical results with regard to the accuracy achieved and the iterations used (this is true for all experiments carried out).
The accuracy required in this experiment is 10⁻⁴.

7.4.2. Solving non-symmetric systems of linear algebraic equations by an orthomin algorithm

The use of iterative methods in the case where the systems solved are neither symmetric nor positive definite is connected with many difficulties. In this case the choice of a proper preconditioner will very often be the most important step in the solution. Some results obtained with a conjugate gradients technique, applied to the residual vectors, will be presented. The particular method used is orthomin (Vinsome [464]). No preconditioning is used (which leads to performing many iterations), but some work is now being carried out to construct a good preconditioner for the method used in this case.

N	NBAND	LINPACK	BLAS 2	DEPTH 0	DEPTH 2	DEPTH 4	DEPTH 8	DEPTH 16
100	21	4	16	45	45	40	39	40
400	41	9	30	79	130	112	114	115
900	61	14	43	91	131	147	157	159
1600	81	18	56	105	135	162	182	190
2500	101	23	66	105	144	174	197	208

Table 7.8

The speeds of computations, in MFLOPS, obtained in solving systems of linear algebraic equations with non-symmetric matrices when the sparsity of the matrix bands is not exploited.

N	NBAND	LINPACK	BLAS 2	DEPTH 0	DEPTH 2	DEPTH 4	DEPTH 8	DEPTH 16
100	21	4	16	33	31	30	30	27
400	41	9	30	77	77	78	78	74
900	61	14	43	97	103	108	109	106
1600	81	18	56	111	118	123	122	121
2500	101	23	66	119	134	136	135	132

Table 7.9

The speeds of computations, in MFLOPS, obtained in solving systems of linear algebraic equations with non-symmetric matrices when the sparsity of the matrix bands is exploited.

N	NBAND	LINPACK	BLAS 2	Sparsity not exploited	Sparsity exploited
100	21	0.01	0.03	0.01 (33.3%)	0.01 (33.3%)
400	41	0.08	0.17	0.04 (23.5%)	0.02 (11.8%)
900	61	0.24	0.51	0.13 (28.1%)	0.04 (7.8%)
1600	81	0.58	1.14	0.32 (28.1%)	0.08 (7.0%)
2500	101	1.13	2.19	0.66 (30.1%)	0.14 (6.4%)

Table 7.10

The computing times, measured in seconds, on AMDAHL VP1100.

The results obtained when unrolling with depth 16 is applied are given in the case where the sparsity is not exploited.

When the sparsity is exploited, the results given in this table are obtained using unrolling with depth 4.

All matrices are generated by using $d = 2.0$.

The percentages are obtained by using the BLAS 2 computing times.

Again the solvers based on calculations by diagonals are compared with direct solvers (the LINPACK, [112], subroutines SGBFA and SGBSL) and with a subroutine based on the same iterative method and the BLAS 2 kernel for performing a matrix-vector multiplication.

The same remarks as in the previous sub-section could be made for the speeds of computations of the "dense" and the "sparse" versions of the code for solving linear systems with banded matrices (in the case where unrolling is not used as well as in the case where unrolling is in use).

It is seen that the LINPACK solver for non-symmetric matrices performs considerably better than the LINPACK solver for symmetric and positive definite matrices. The same is true when the speeds of computations of the BLAS 2 solvers (for non-symmetric matrices and for symmetric and positive definite matrices) are compared. This and the fact that many more iterations are needed now to obtain the accuracy required show why the computing times obtained when the calculations are organized by diagonals are not so good as those these obtained in the previous section. As already mentioned, some kind of preconditioning seems to be necessary in this situation.

It should be mentioned here that the LINPACK solver will perform much better than the iterative solvers on a scalar machine.

7.4.3. Special matrices

The improvement achieved by unrolling is caused by the reduction of the sweeps in the outer loops. Of course, it will be even better if the double loops can be replaced by simple loops, which can sometimes be done. In many applications the matrices involved in the computations are very special: the number of their non-zero diagonals is very small (i.e. three or five). In such a case the codes used to obtain the results in the previous two sections can be simplified so that all double loops are replaced by single loops. The implementation of this idea is straightforward. Kernels prepared in this way have been applied in some of the models at the National Environmental Research Institute. The numerical results indicate that such kernels are very useful, especially when the problems treated are very large.

7.4.4. Some general remarks on the use of iterative methods on vector processors

The storage scheme should be selected carefully when a vector processor is to be used in the iterative solution of large systems of linear algebraic equations with banded matrices. The matrix elements should be stored by diagonals and all computations should be organized by diagonals in order to utilize completely the power of the vector machine used.

The use of iterative methods in the solution of linear algebraic equations could result in many (or even too many) iterations. Therefore preconditioning should be used when this happens. If a vector processor is in use, then not only should the number of iterations be decreased by applying the preconditioner chosen, but it is also desirable to keep the high speed of computations. These two requirements work in opposite directions (as often happens in practice) and, therefore a compromise must be found. The search for a good preconditioner, especially for large systems of linear algebraic equations with general banded matrices, is continuing at the National Environmental Research Institute.

Solving non-symmetric LAE's by iterative methods is an important problem. Algorithms based on a minimization of the residual vectors by conjugate gradients techniques have been used until now. The results that are obtained with such algorithms are rather encouraging (see also the tables presented in this sub-section). There are plans to try also GMRES from Saad and Schultz [379], as well as some other methods.

7.4.5. Using direct methods on vector processors

Assume that systems of LAE's whose matrices are banded are to be solved on vector processor. The speed of computations is usually not very fast when direct methods are used in the solution of such systems. This has been demonstrated in the previous sub-sections of this section by using appropriate subroutines of package LINPACK (see Dongarra et al. [112]). The main reason is that the loops that are involved in the computations are short (this fact has already been discussed). However, direct methods can be very useful in some situations.

There are at least two important cases in which the direct methods may be preferable when the choice must be made between iterative methods and direct methods). The first case is connected with the application of finite elements in the space discretization of certain partial differential equations. This leads to the solution of large systems of ODE's of the following type:

$$(7.6) \quad P \, dg/dt = Hg + f,$$

where P is a constant matrix (very often also symmetric and positive definite), H is a variable matrix that is dependent, for example, on the time variable, while both g and f are vectors. Assume first that the time-integration method is implicit. Then the discretization of (7.6) leads to the solution of many systems of LAE's whose coefficient matrices are of the form:

$$(7.7) \quad J = P + \Delta t \, \beta \, H,$$

which means that these matrices vary from one time-step to another. In this situation iterative methods may be more profitable than direct methods. However, the situation changes when explicit time-integration methods are used in the numerical treatment of (7.6). In the latter case again many systems of LAE's are to be solved, but these systems have the same coefficient matrix P , which is assumed to be constant. This means that matrix P can be factorized in the beginning of the time-integration and then the triangular factors (an upper and a lower triangular matrices) can be used repeatedly during the whole integration process. The direct methods are useful in this situation. The speed of computations is again low when vector processors are used but the number of computations is reduced dramatically.

Another important case where direct methods are preferable is the case where the iterative methods are slowly convergent. In this situation the high speed of computations will not be sufficient to compensate for the increased number of computations because too many iterations have to be performed.

These two cases (and other cases can be found), in which it may be better to choose direct methods, indicate that one must have both iterative methods and direct methods in the software. The computations in the iterative methods consist of basic linear algebra operations which as a rule vectorize very well. The direct methods with banded matrices do not vectorize well, but the number of simple arithmetic operations can be considerably smaller than the corresponding number for the iterative methods.

Thus, the choice depends both on the particular problem that is to be solved and on the computer that is to be used in the computations. As a rule, the iterative methods perform better when powerful vector processors are used. There are both direct methods and iterative methods included in the software that is currently used to treat numerically **the Danish Eulerian Model**.

7.5. Vectorizing fast Fourier transforms

When the pseudospectral algorithm described in Section 3.2 is used to discretize the space derivatives that are involved in (7.1), Fourier transforms are to be used to determine the coefficients in the trigonometric interpolation polynomials (3.17) and (3.19). The number of arithmetic operations needed to calculate these coefficients is reduced considerably by using the so-called Fast Fourier Transforms (FFT's). Nevertheless, the determination of these coefficients remains one of the most time-consuming parts in the numerical treatment of an air pollution model when the pseudospectral algorithm is used. Therefore it is important to perform this part of the computations efficiently. This is also true when vector computers are to be used.

Assume now that a simple version of the model (the sulphur model with two pollutants only) is used. Assume also that this model is considered as a three dimensional model. Then the space discretization part takes approximately 85% of the CPU time needed to perform a one-year run on CRAY 1 (see Zlatev [520]). This is why the FFT's must be vectorized on a high level. Fortunately, the FFT's are basic operations for which one can easily find

high quality software. Two packages were used in the experiments with the three dimensional sulphur model (Zlatev [520]). The first one was developed by Swarztrauber (see, Zlatev [520]). The second one was developed by Temperton (see again, Zlatev [520]) and implemented, as subroutine FFT991, at the European Centre for Medium-range Weather Forecasts (ECMWF) in Reading, England. Results that are obtained by these subroutines when the simple two-dimensional rotation test (which has been described in Section 4.1) is used are shown in Table 7.11.

Computer used	FFT algorithm	Computing time
CDC Cyber 175	Swarztrauber's FFT's	2572
CRAY X-MP48	Swarztrauber's FFT's	298
CRAY X-MP48	FFT991 at ECMWF (Reading)	25

Table 7.11
Performance of different FFT's in a run of the rotation test
of the advection module of the model (see Chapter 4).
The number of ODE's treated at each time-step is 1024.
12500 time-steps were completed to achieve 50 rotations
of the non-trivial part of the initial distribution.

The results in Table 7.11 demonstrate clearly the importance of proper vectorization of the FFT's. A good vectorization of the subroutines that perform FFT's leads to a very substantial improvement of the speed of computations. This is especially true for the subroutine FFT991, which also performs the so-called multiple FFT's. The use of multiple FFT's is very efficient when vector and/or parallel computers are in use.

The results show also that the old-fashioned sequential computers (in this case represented by a CDC Cyber 175) are not competitive at all with the modern vector processors when the code is well vectorized.

It should be mentioned here that the subroutine FFT991 is partly written in machine language and therefore can be used only on CRAY products. However, the situation has been improved after 1988, because subroutines based on the Temperton algorithm were included as standard subroutines in the well-known NAG Library [229] (the subroutines of the NAG Library [229] are installed in many computing centres). The NAG FFT subroutines were tested in Zlatev [520]. The performance of the NAG FFT subroutines is quite similar to that of the Temperton's subroutine FFT991.

7.6. Concluding remarks and references concerning efficient runs of large air pollution models on vector machines

Vector computers are now commonly used in the computations with large LRTAP models. These computers are fast. Their compilers become more and more intelligent (and, therefore, more and more operations involved in the codes are automatically optimized even when the programmers have not coded them in the best way). However, some more complicated operations are to be treated carefully. This is especially true (see [520]-[524], [526], [532], [540], [547]) for

- (i) basic linear algebraic operations that involve banded matrices,
- (ii) solution of systems of linear algebraic equations whose coefficient matrices are banded,
- (iii) computations with Fast Fourier Transforms

and

- (v) the chemical reactions.

The separation of the most time-consuming parts of a large LRTAP model is discussed in **Section 7.1**. It has been pointed out there that the operations involving banded matrices and Fast Fourier Transforms must be treated carefully on a vector processor, because it is not easy to achieve high speed of computations for these two types of operations. On the other hand, the proper vectorization of these operations improves considerably the total performance of a large LRTAP model.

The basic linear algebra operations play an important role when finite differences or finite elements are used in the discretization of the space derivatives of an LRTAP model. Most of these operations are well-understood now (and efficient subroutines exist at many computing centres). This is especially true when dense matrices are involved in the basic linear algebra operations. However, the situation changes when banded matrices appear in the computations. The use of vectorized basic linear algebra operations in connection with banded matrices is briefly described in **Section 7.2**; see also [468], [469], [520] and [547].

Among the basic linear algebra operations, the matrix-vector multiplication is one of the most important. It is highly desirable to perform this operation efficiently. Algorithms for vectorized matrix-vector multipliers are developed in **Section 7.3**; they are based on ideas described in [523] and [547]. The vectorizable algorithms in the case where the matrix is banded are studied in §7.3.1. A special class of banded test-matrices is introduced in §7.3.2, and the efficiency of the proposed algorithms is demonstrated (by using these test-matrices) in §7.3.3. Special algorithms for unrolling double loops and exploiting the sparsity within the matrix band are presented in §7.3.4 and §7.2.5 respectively. Finally, it is shown in §7.3.6 that the algorithms can efficiently be used in multitasking procedures.

The solution of systems of linear algebraic equations whose matrices are banded is studied in **Section 7.4**. First iterative methods for symmetric and positive matrices are studied in §7.4.1 (further discussion of these methods can be found in [523]). Iterative methods for non-symmetric matrices are considered in §7.4.2 (see also [524]). The methods described in §7.4.1 and in §7.4.2 can be implemented even more efficiently for some special matrices; this possibility is discussed briefly in §7.4.3. Some common remarks concerning the use of iterative methods in the solution of systems of linear algebraic equations with banded matrices are given in §7.4.4. The use of direct methods (many such methods are discussed in [170], [171], [244], [438], [439], [444],[465], [473], [482]-[489]) in the solution of systems of linear algebraic equations with banded matrices may sometimes be more preferable than the use of iterative methods; this issue is discussed briefly in §7.4.5. Estimates of the condition numbers of the matrices involved in the computations may be very useful in the choice of a numerical method (iterative or direct). Estimators of the condition number of a matrix are studied in [238], [239], [322], [559]. It should also be mentioned here that condition number estimators are available in many modern solvers for systems of linear algebraic equations as, for example, in many solvers in the packages LINPACK [112] and LAPACK [8].

Fast Fourier Transforms (FFT's) are also often used when large LRTAP models are to be handled on vector processors. The FFT's are very time-consuming, and therefore it is crucial to perform them efficiently. The use of vectorizable FFT's is discussed in **Section 7.5**. The efficiency of the vectorized FFT's, and especially the vectorized multiple FFT's, is illustrated by numerical examples (these numerical methods are also studied in [520], [522], [532] and [540]).

Vectorizable algorithms for the chemical sub-model have not been described in this chapter. This important problem will be discussed in the following chapter.

CHAPTER 8

RUNNING MODELS ON PARALLEL MACHINES WITH SHARED MEMORY

"The term "computational science" was first used by Ken Wilson (awarded a Nobel Prize for his work in physics) to refer to those activities in science and engineering that exploit computing as their main tool. Indeed, in the past 10 years, computing has become an essential tool for advancing various science and engineering disciplines. Moreover, with the advent of parallel computers, it become clear that parallel computing has become an integral part of many aspects of these disciplines"

Ahmed Sameh and John Riganati: "Computational Science & Engineering", IEEE Computer, Vol 26, No. 10, 1993, 9-12.

Parallel computers with shared memory are rather popular. The main reason for this popularity is the fact that it is relatively easy to achieve high speeds of computations when such computers are used. Furthermore, and this is in many cases even more important, good performance could often be achieved with minimal efforts. Therefore one should consider the possibility of running large air pollution codes on parallel machines with shared memory. It must be emphasized here that many companies that originally produced single vector processors are now producing parallel computers with shared memory. CRAY is a typical representative of this group of computer companies.

Several modules of **the Danish Eulerian Model** (in fact, some of the test-problems that were introduced in Chapter 4) were run on several parallel computers with shared memory (ALLIANT FX/80, ALLIANT FX/2800, CONVEX and CRAY Y-MP8). The most important physical processes involved in the model, the advection and the chemistry, are represented in these test-problems. It seems clear that if these test-problems can be run efficiently on parallel machines with shared memory, then the whole **Danish Eulerian Model** (and also many other large air pollution models) can also be run efficiently on this kind of parallel computers.

It will be demonstrated in this chapter that efficient numerical methods¹ for running air pollution models on parallel computers with shared memory can be selected and implemented. The most important parts of the work that has to be performed in order to optimize a large application code (not necessarily an air pollution code) will also be discussed. If the processors of the parallel computer that is to be used have a vector capability, then the optimization process should be carried out on two levels:

- (i) vectorization on every processor

and

- (ii) parallelization across the processors.

¹ In the context of this chapter, a numerical method is efficient if it is both sufficiently fast and sufficiently accurate.

Some numerical results that are obtained on two ALLIANT computers (ALLIANT FX/80 and ALLIANT FX/2800) will be presented. However, it should be emphasized that modules of **the Danish Eulerian Model** were also run on several other parallel computers with shared memory.

8.1. Identifying operations that can be run concurrently

If a large application code (not necessarily an LRTAP model) is to be run on a parallel computer with shared memory, then first and foremost the parts of the codes that can be run in parallel should be identified. Of course, one is not interested in all parts of the codes that can be run concurrently. The task of finding all parts of a large application code that can be run in parallel can be very difficult. Therefore, the parts of the code that are less time consuming should, in the first phase at least, not be considered for parallelization.

This means that one should start the investigation with the most expensive (the most time consuming) parts of the code, and the first task should be to try to parallelize these most expensive parts of the code. Once the parallelization of the most time-consuming parts of the code has successfully been completed, one should investigate once again carefully the performance of the other parts of the code. This is necessary because the parts of the code that have been ignored in the starting phase (were not among the most time-consuming parts of the code in the very beginning of the optimization process) could become relatively expensive after the parallelization of the most time-consuming parts of the code. If this happens, then one should proceed by trying to parallelize the parts that have become relatively expensive after the parallelization of the most time-consuming parts of the code. It could be necessary to repeat this procedure several times. It should also be emphasized here that the procedure sketched above is well-known and commonly used when big application codes are to be optimized on parallel computers.

In many cases the identification of the parts of a large application code that can be run in parallel is not sufficient. This is especially true when the parallel computer under consideration has a cache memory. The access of data that are in the cache memory of the computer is as a rule much faster than the access of data from the global memory. Therefore it is worthwhile to work as long as possible with data from the cache memory. Unfortunately, in most of the cases there is no direct way to decide what data must be in cache (the compiler decides this). Nevertheless, one should try to organize the computations so that the same data are used as long as possible (hoping that these data will stay in the cache memory as long as they are needed).

8.2. Calling subroutines in concurrent runs

Consider again the case where the air pollution model is a two-dimensional model (the three-dimensional case leads to an increase in the amount of the computational work, but the same ideas can be applied also when the model is three-dimensional).

As mentioned several times in the previous chapters, the most time-consuming parts of an LRTAP code are the advection sub-model and the chemical sub-model. Therefore the parallelization of these two sub-models will be described in some more detail (§ 8.2.1 and § 8.2.2). If a splitting procedure is applied, then the advection of every chemical compound can be carried in parallel, while the chemical reactions at every grid-point of the discretized space domain can be performed concurrently. The other two processes, diffusion and deposition, will be discussed in § 8.2.3. The parallelization of three-dimensional models will be sketched in § 8.2.4, while some short remarks will be given in § 8.2.5.

8.2.1. Parallelization of the advection part

Let us assume that we have a subroutine `ADVECT` that could perform one advection time-step for one of the pollutants involved in the model. This means that the subroutine `ADVECT` performs separately one time-integration step for each equation of the advection sub-model (see more details in Section 5.2):

$$(8.1) \quad \partial c_s^{(1)} / \partial t = - \partial (u c_s^{(1)}) / \partial x - \partial (v c_s^{(1)}) / \partial y$$

In a call of subroutine `ADVECT` the integration is performed from some t_k to $t_k + \Delta t$, where Δt is the time-stepsize chosen. The computations during the advection time-step k ($k=1,2,\dots$) can be carried out by using the loop given in Fig. 8.1.

```

DO 10 I=1,NSPECIES
  CALL ADVECT
10 CONTINUE
```

Figure 8.1

**The loop that can be used to perform an advection time-step
(`NSPECIES` is the number of pollutants involved in the model).**

If the air pollution code is run on a parallel computer with shared memory and if it is possible to call subroutines in a concurrent mode (on nearly all such computers this is possible), then the loop shown in Fig. 8.1 can be run in parallel.

If the number of pollutants `NSPECIES` is equal to the number of processors or if `NSPECIES` is a multiple of the number of processors, then it will be very efficient to call concurrently subroutine `ADVECT`. The reason for the high efficiency is clear: every processor has received a task, and moreover the amount of work that has to be done by any processor is equal to the amount of work that has to be done by any of the other processors. In other words, every processor will be busy at the start of the k 'th advection time-step and will likely be busy until the other processors are busy (which is a perfect situation when parallel computers are used).

However, if the above condition is not satisfied, then the computational process will not, in general, be very efficient. Assume, for example, that there are four pollutants, while the number of processors is 40. Then only four processors will participate in the computations during the integration of the advection sub-model (i.e. only 10% of the computer power of the parallel machine will be exploited). This is clearly not satisfactory, and **the parallelization of the advection sub-model has to be carried out on a lower level in this situation**. If such an action is properly performed, then the efficiency will strongly depend on the numerical methods chosen.

If finite elements are used in the discretization of the spatial derivatives of (8.1), then the calculations with banded matrices are important (see the previous chapter). It is not very difficult to obtain a good parallel performance in the computations with banded matrices by using the techniques that has been described in the previous chapter (in fact, one example has been given there; see § 7.3.6).

If the pseudospectral algorithm that has been described in Chapter 3 is used, then the subroutines that perform FFT's (Fast Fourier Transforms), which are rather expensive, might be called in parallel.

Parallel computations can also be organized when other numerical methods are used. For example, if finite differences are used, then the situation will be rather similar to the case where the advection part is discretized by using finite elements.

It is important to emphasize here that if it is not efficient to parallelize the loop given in Fig. 8.1, then it is relatively easy, when LRTAP models are treated on parallel machines, to achieve parallel computations at some lower level. However, while it is not difficult to achieve parallel computations at different levels, it might be very difficult to find out the optimal level at which the parallelization is to be carried out. The solution of this task may also depend on the solution of the tasks of obtaining parallel computations in the other sub-models of the air pollution model under consideration.

8.2.2. Parallelization of the chemical part

The chemical sub-model is very often the most time-consuming part of the computations when any LRTAP model is run on a computer. Therefore it is crucial to obtain high performance when the chemical sub-model is executed. It has been pointed out, in Chapter 3,

that the chemical reactions will normally lead to very stiff systems of ODE's. This imposes strong requirements for efficiency on the numerical methods that are to be used in the chemical part of a large air pollution model. The problem of finding an efficient numerical method for the chemical part is still an open problem (see again Chapter 3). However, if a numerical method for the chemical sub-model is somehow chosen, then it is relatively easy to achieve high computational speeds when the chemical part is run on many high speed computers (this fact has been mentioned in the previous chapter). In this paragraph it will be described how high computational speeds can be achieved when parallel computers with shared memory are used.

Let us consider the system of ODE's by which the chemical part of a large LRTAP model is described (see Chapter 3 for more details about the chemical reactions and the systems of ODE's by which the chemical reactions are expressed in air pollution models):

$$(8.2) \quad \frac{dc_s}{dt} = P_s - L_s C_s,$$

where $s=1,2,\dots,q$ and q is the number of pollutants involved in the model, while $P_s = P_s(c_1, c_2, \dots, c_q)$ and $L_s = L_s(c_1, c_2, \dots, c_q)$ are nonlinear terms that are typical for the chemical reactions. These terms describe the production and the loss rate for the pollutant under consideration during the chemical processes in which it is involved.

The system defined by (8.2) contains $NX*NY*NZ*NSPECIES$ equations (where NX is the number of grid-points on the grid-lines parallel to the Ox axis, NY is the number of grid-points on the grid-lines parallel to the Oy axis, NZ is the number of grid-points on the grid-lines parallel to the Oz axis, and $NSPECIES=q$ is the number of pollutants involved in the air pollution model). However, the structure of the system (8.2) is very special. The air pollutants in a given grid-point in the space domain may interact chemically with each other, but they do not interact chemically with pollutants from the other grid-points in the space domain. This means that the big system of ODE's (8.2) is split into $NX*NY*NZ$ smaller systems, where $NX*NY*NZ$ is the total number of grid-points in the space domain of the air pollution model under consideration. **Each of these smaller systems contain only $NSPECIES$ equations and can be treated independently from the other small systems.** The i 'th small ODE system, $i=1,2,\dots,NX*NY*NZ$, can be written as follows:

$$(8.3) \quad \frac{dc_{si}}{dt} = P_{si} - L_{si} C_{si}.$$

Let us assume again, as in the previous paragraph, that a subroutine CHEMI, which performs one time-step in the numerical integration of any of the small systems (8.3), is prepared. This means that the subroutine CHEMI can perform the integration of the chemical sub-model (at any of the spatial grid-points) from any time grid-point t_k to the next time grid-point $t_k + \Delta t$, where $k=1,2,\dots$ and Δt is the time-stepsize chosen. It is important to mention here that the time-stepsize used in the chemical part of an air pollution model is not necessarily the same as the time-stepsize used in its advection part. In fact, very

often the time-stepsize used in the chemical part is considerably smaller than that used in the advection part. The chemical time-stepsize used in **the Danish Eulerian Model** is six times smaller than the time-stepsize that is used in the advection part of the same model, see Zlatev et al. [538,539]. As a rule, the chemical part must be integrated by a smaller time-stepsize because the systems of ODE's (8.3) are very stiff.

The module **CHEMI** is often (but not always) very useful when a large air pollution model is to be run on parallel computers. The computations can be organized in a loop as shown in Fig. 8.2 when such a module is available.

```
DO 20 I=1,NX*NY*NZ  
  CALL CHEMI  
20 CONTINUE
```

Figure 8.2
The loop that can be used to perform a chemical time-step
($NX*NY*NZ$ is the number of spatial grid-points).

The loop in Fig. 8.2 is very similar to the loop in Fig. 8.1. However, there is one difference which is essential on many parallel computers with shared memory. The parameter **NSPECIES** is normally not very large (the number of pollutants is 35 in **the Danish Eulerian Model**). This could cause some difficulties when the number of processors is greater than **NSPECIES**, and force the developers of efficient air pollution codes to try to obtain parallel computations on a lower level (see the discussion after Fig. 8.1 in § 8.2.1). The parameter used in the loop in Fig. 8.2, $NX*NY*NZ$, is normally much greater than the number of processors in a parallel computer with shared memory (this is true even when the model is two-dimensional; in the new version of **the Danish Eulerian Model**, with $NX=NY=96$ and $NZ=1$, the value of this parameter is 9216). This means that

while one may have problems with the number of parallel tasks when the loop in Fig. 8.1 is used (the number of tasks may be less than the number of processors), problems with the number of parallel tasks will normally not appear when the loop in Fig. 8.2 is used in order to parallelize the chemical part of a large air pollution model.

However, this does not mean that the loop used in Fig. 8.2 is always perfect. Some problems may arise when each of the processors of a parallel computer with shared memory has a vector capability. Since the processors of many parallel computers with shared memory have a vector capability, this situation is important, and therefore it deserves special attention; the situation where both vectorization and parallelization of the computational process is needed will be studied in the next section.

8.2.3. Parallelization of the other parts

Assume that a two-dimensional air pollution model is to be handled. The two parts of the model that have not been discussed in the preceding two paragraphs, the diffusion part and the deposition part, can be optimized (for efficient runs on parallel computers with shared memory) in a similar way as the advection part and the chemical part. The main ideas that can be applied in the optimization process are described briefly below.

Consider first the diffusion sub-model. Assume that a subroutine DIFFUS, which performs one time-step for one of the equations in the diffusion sub-model, has been developed. If subroutines can be called concurrently on the computer that is available, then parallel computations can be achieved by using the code given in Fig. 8.3.

This code is indeed very similar to the code given in Fig. 8.1. Therefore all conclusions made in § 8.2.1 for the advection sub-model are also valid for the diffusion sub-model (and it is not necessary to repeat these conclusions here).

```
DO 30 I=1,NSPECIES  
  CALL DIFFUS  
30 CONTINUE
```

Figure 8.3

**The loop that can be used to perform an advection time-step
(NSPECIES is the number of pollutants involved in the model).**

On some parallel computers with shared memory it is worthwhile to have a few parallel tasks only, but it is required that each task is as big as possible. On such parallel computers it would be more efficient to combine the loop given in Fig. 8.1 with the loop given in Fig. 8.3 than to run these two loops separately. Such a combined loop is shown in Fig. 8.4.

```
DO 10 I=1,NSPECIES  
  CALL ADVECT  
  CALL DIFFUS  
10 CONTINUE
```

Figure 8.4

**The loop that can be used to perform an advection-diffusion time-step
(the number of tasks is the same as that in Fig. 8.1 and Fig. 8.3,
but the tasks are bigger).**

On some parallel computers with shared memory the number of parallel tasks may be insufficient (when the number of processors is greater than the number, NSPECIES, of parallel tasks). It has been pointed out in § 8.2.1 that one has to try to parallelize the code on a lower level when this is the case. If the combination advection-diffusion is considered, then one can try to avoid the attempts to parallelize on the lower level (which may be difficult) by pipelining the advection and the diffusion parallel tasks. An example is shown in Fig. 8.5. It is seen that the number of parallel tasks is increased substantially, by a factor of two, when pipelining of the advection and diffusion parallel tasks has been implemented in the code. It should be noted here that slight modifications of the numerical algorithms will be needed when such a pipelining procedure is to be implemented.

Consider now the deposition sub-model. Assume that a subroutine DEPOS, by which one deposition time-step can be performed, has been developed. Since the deposition of the pollutants in a given spatial grid-point does not depend on the concentrations of the pollutants in any of the other spatial grid-points, the same ideas as those used in the chemical sub-model can be applied to optimize the deposition sub-model for runs on parallel computers with shared memory. In other words, a loop that is quite similar to the loop from Fig. 8.2 can be applied (and, moreover, all conclusions that have been drawn in § 8.2.2 are also valid for the deposition sub-model). Such a loop is shown in Fig. 8.6.

```

DO 10 I=1,2*NSPECIES
  IF(I.LE.NSPECIES) THEN
    CALL ADVECT
  ELSE
    CALL DIFFUS
  END IF
10 CONTINUE

```

Figure 8.5

The loop that can be used to perform an advection-diffusion time-step by pipelining the advection and the diffusion parallel tasks (the number of tasks is twice greater than that in Fig. 8.1 and Fig. 8.3).

```

DO 40 I=1,NX*NY*NZ
  CALL DEPOS
40 CONTINUE

```

Figure 8.6

The loop that can be used to perform a deposition time-step (NX*NY*NZ is the number of spatial grid-points).

It is again possible, as for the advection-diffusion case, both to enlarge the parallel tasks by combining the deposition time-step with the chemical time-step (see Fig. 8.7) and to pipeline the deposition tasks with the chemical tasks (see Fig. 8.8). In the former case the number of parallel tasks is unchanged, but the tasks become bigger (this could be useful on many parallel computers with shared memory). In the latter case the number of tasks becomes twice greater, but their sizes are unchanged. In many cases it will probably not be necessary to increase the number of parallel tasks in the deposition and chemistry sub-models because, as emphasized in § 8.2.2, the number of tasks is normally very large both in the loop from Fig. 8.2 and in the loop from Fig. 8.6. It should be reiterated here that slight modifications of the numerical algorithms will be needed when such a pipelining procedure is used.

```
DO 20 I=1,NX*NY*NZ
  CALL DEPOS
  CALL CHEMI
20 CONTINUE
```

Figure 8.7

**The loop that can be used to perform a deposition-chemistry time-step
(the number of tasks is the same as that in Fig. 8.2 and Fig. 8.6,
but the tasks are bigger).**

```
DO 20 I=1,2*NX*NY*NZ
  IF(I.LE.NX*NY*NZ) THEN
    CALL DEPOS
  ELSE
    CALL CHEMI
  END IF
20 CONTINUE
```

Figure 8.8

**The loop that can be used to perform a deposition-chemistry time-step
by pipelining the advection and the diffusion parallel tasks
(the number of tasks is twice greater than that in Fig. 8.2 and Fig. 8.6).**

It should again be noted here that in many places in this book an assumption has been made that the deposition process can be described by simple linear terms (see Chapter 2 and Chapter 3). Such an assumption is not needed when concurrency is discussed (the only thing that is important in order to parallelize the deposition sub-model is to prepare a module, the subroutine `DEPOS`, that executes one time-step for each pollutant of the deposition sub-model.

Similar remarks, as those concerning subroutine `DEPOS`, are also valid for the other processes (the advection, the diffusion and the chemistry). The particular physical procedures

that are applied in the different parts of the air pollution models and/or the particular numerical methods that are selected during the execution on a computer **do not affect the parallelization techniques**. The computations described in the loops given in Fig. 8.1 - Fig. 8.8 will be carried out **in parallel** for any physical procedure and for any numerical method (but, of course, the computing times will depend both on the physical procedure applied and on the numerical method chosen). These remarks show that the ideas discussed in this chapter are rather general.

8.2.4. Parallelization of three-dimensional models

The work with three-dimensional models causes a lot of difficulties (very large sets of input data, very large systems of ODE's that are to be solved at every time-step, many time-steps, etc.). On the other hand, it becomes easier to run different parts of the code in parallel when the code arises from a three-dimensional air pollution model. The main ideas that can be applied in order to obtain parallel computations remain in principle the same (as those described in the previous paragraphs of this section). However, the degrees of freedom are in general increased. This means it becomes relatively easier both to create more parallel tasks and to create bigger tasks when a three-dimensional air pollution model is handled. On the other hand, the computing work is increased very substantially when the model is three dimensional.

The fact that **the same ideas** can be exploited to parallelize both two-dimensional air pollution models and three-dimensional models has already been demonstrated in § 8.2.2 and in § 8.2.3 during the discussion of the parallelization of the chemical and the deposition sub-models. Indeed, the parallelization techniques presented there for these two sub-models are valid both for two-dimensional models (with $NZ = 1$) and for three-dimensional models (with $NZ > 1$).

The fact that **the degrees of freedom are increased** can be demonstrated as follows. Consider the horizontal advection sub-model. Assume that for each air pollutant the subroutine ADVECT performs one time-step per one layer. Then it is clear that the code given in Fig. 8.1 can be rewritten as shown in Fig. 8.9.

```

DO 10 I=1,NZ*NSPECIES
  CALL ADVECT
10 CONTINUE

```

Figure 8.9

The loop that can be used to perform an advection time-step (NSPECIES is the number of pollutants involved in the model).

It is seen (by comparing Fig. 8.1 with Fig. 8.9) that the number of parallel tasks is increased by a factor of NZ . It has been mentioned in § 8.2.1 that the number of parallel tasks in the loop from Fig. 8.1 can sometimes be smaller than the number of processors for some parallel computers. It is clear that this disadvantage will very often disappear when the air pollution model under consideration is three-dimensional.

The horizontal diffusion sub-model can be treated in a very similar way as the advection sub-model. The parallel tasks can be enlarged by combining the advection tasks with the diffusion tasks and/or by combining the chemistry tasks with the deposition tasks. The parallel tasks can be pipelined (advection tasks with diffusion tasks and/or chemistry tasks with deposition tasks). All these procedures can be carried out in a similar way as the corresponding procedures in § 8.2.3.

The only process that is specific for a three-dimensional model is the vertical advection-diffusion. This process is described by the following system of PDE's (see Chapter 3):

$$(8.4) \quad \partial c_s^{(5)} / \partial t = - \partial (w c_s^{(5)}) / \partial z + \partial (K_z \partial c_s^{(5)} / \partial z) / \partial z,$$

where w and K_z are the vertical wind velocity and the vertical diffusion coefficient respectively, while $s=1,2,\dots,q$ (NSPECIES is used instead of q in this chapter in order to achieve compatibility with some of the agreements in the FORTRAN language). Assume that linear one-dimensional finite elements are applied in the discretization of (8.4) on each vertical grid line and for each pollutant. Then the system of ODE's (8.4) will be reduced to $NX*NY*NSPECIES$ smaller systems of ODE's. Each of these systems can be written in the following form (see again Chapter 3):

$$(8.5) \quad P_i dg_i / dt = H_i g_i + f_i,$$

$$P_i \in \mathbb{R}^{NZ \times NZ}, \quad H_i \in \mathbb{R}^{NZ \times NZ},$$

$$dg_i / dt \in \mathbb{R}^{NZ}, \quad g_i \in \mathbb{R}^{NZ}, \quad f_i \in \mathbb{R}^{NZ};$$

where the notation used can be explained as follows:

(i) $i \in \{1,2,\dots, NX*NY*NSPECIES\}$,

(ii) P_i is a symmetric and positive definite matrix which depends only on the finite elements chosen,

(iii) H_i is a matrix that depends on the wind velocities as well as on the diffusion coefficient K_z (and, therefore, matrix H_i is in general time-dependent),

(vi) $g_i = g_i(t)$ is a vector that contains the values of the concentrations at the grid-points along the vertical grid-line under consideration and at time t ,

(v) $dg_i/dt = dg_i(t)/dt$ is a vector whose components are the time derivatives of the concentrations that are stored in vector $g_i(t)$,

(vi) $f_i = f_i(t)$ is a vector whose components depend on the boundary condition used in the definition of the problem (it will be assumed here that Dirichlet boundary conditions are always specified; i.e., the values of the concentrations are given on the boundaries, but this assumption is in fact not needed).

Assume that a subroutine VERTIC that performs one time-step in the integration of one equation of the system of ODE's (8.5) has been developed. Then the computational process can be organized by using the loop shown in Fig. 8.10.

```

DO 50 I=1,NX*NY*NSPECIES
  CALL VERTIC
50 CONTINUE

```

Figure 8.10

**The loop that can be used to perform
one time-step along the vertical grid-lines
(NX*NY*NSPECIES is the number of parallel tasks
in this part of the computational work).**

The main conclusion, which can be drawn from the discussion in this paragraph, is that in general it becomes easier to design parallel tasks when the model is three-dimensional. It should be emphasized that parallel computations in this situation are also highly desirable because the three-dimensional models are very expensive in terms of computing time (and also in terms of storage, but the storage requirements become less and less important on the big modern architectures). In fact, a large three-dimensional air pollution code can be handled successfully on a computer only if the following three conditions are satisfied:

(i) the computer that is available is a very fast parallel and/or vector architecture,

(ii) the numerical algorithms are carefully selected (so that they are suitable for the computer that is to be used

and

(iii) the code is optimized so that the great potential power of the computer is exploited as well as possible (this implies that if the computer is parallel, then the most time-consuming parts of the code must be run in parallel).

8.2.5. Some remarks

Several techniques by which parallel tasks can be determined have been described in the previous paragraphs of this section. The possibilities of exploiting some special features of the parallel machines (as, for example, the optimal use of the cache) were not discussed in detail. There are several reasons for this preference of discussing mainly the parallel tasks.

The most important reason can be explained as follows. The parallel tasks depend on the problem solved and can be executed in parallel on any parallel computer. The exploitation of the special features of the computers depends on the particular parallel computer that is available. The development in this field is very quick, and features that are very important now (and could without any doubt be exploited to improve the performance of the code very considerably) may become useless for the products that will dominate the market in a couple of years.

On the other hand, some of the special features are typical for rather large classes of parallel machines with shared memory. Two important examples are related to the size and the number of the parallel tasks. Big parallel tasks are performed more efficiently than small parallel tasks on many parallel machines with shared memory. The number of parallel tasks becomes important when the number of processors is large. An attempt to discuss briefly some special features that are typical for many parallel computers has been made.

If the parallel computer that is to be used is given, then one should try to exploit its special features as much as possible. This is especially true if large three-dimensional models (not necessarily air pollution models) are to be treated. Our belief is that if the parallel tasks are already identified, then it would not be very difficult to exploit the special features of the particular parallel computer that is to be used (of course, while such a statement is in general true, it is also true that one can find examples where it is very difficult to exploit in an efficient way the special features of a given parallel computer).

The results presented in this section illustrate the important fact that a good modulation of the code is a rather powerful tool in the efforts to obtain a good performance on high-speed computers. Note that this last remark is valid not only when large air pollution codes are treated on parallel computers, but also when any large application code is to be optimized for runs on parallel computers.

8.3. Performing both vectorization and parallelization

Assume that the processors of a parallel computer with shared memory have a vector capability (some vectorization techniques have been described in the previous chapter). This is often the case; CRAY, CONVEX and ALLIANT are among the companies that have such products. In this situation it is important to solve successfully the following two major problems in the efforts to achieve high-speed computations:

- (i) to carry out the computations on each processor in a vector mode

and

- (ii) to parallelize the computations across the processors and to keep the processors busy as long as possible.

The solution of these two important tasks (in the case where large air pollution models are run on parallel computers with shared memory whose processors have a vector capability) will shortly be discussed in this section. The main purpose will be to illustrate the common ideas that can be used, and not to go into details. The common ideas that can be used successfully to get parallel computations are rather general and can be applied for any code on any parallel computer. On the other hand, the details depend strongly both on the particular code and on the particular computer. The hope is that, when the main ideas are correctly implemented, the developer can go one step further and look at the particular features of both the code and the computer available in an attempt to improve the performance.

8.3.1. Parallel-vector computations in the advection sub-model

The case, where the loop given in Fig. 8.1 can efficiently be used (i.e., if the number of air pollutants is either equal to the number of processors or the number of pollutants is a multiple of the number of processors), is relatively easy to treat. Indeed, if this is so, then the subroutine `ADVECT` is called in parallel and executed (perhaps several times at every time-step) by each processor. Therefore the only thing that is to be done is to vectorize the computations in subroutine `ADVECT`. This should not cause problems; the techniques proposed in the previous chapter can be used to vectorize subroutine `ADVECT`. It should be stressed here that it is efficient to use the loop given in Fig. 8.1 on many parallel computers with shared memory.

The case where the loop given in Fig. 8.1 cannot be used efficiently on the parallel computer at our disposal (for example, because the number of processors is greater than parameter `NSPECIES`) is much more difficult. In this situation one should try to find the most time-consuming parts in subroutine `ADVECT` and then one should prepare subroutines by which these parts can be run concurrently. Moreover, one should try to vectorize each of these subroutines. This means that the same ideas as those described in § 8.2.1 can be used,

but on a lower level. An example for parallel-vector computations during the execution of the advection sub-model on a CRAY computer has been discussed in the previous chapter (see § 7.3.6).

A crucial question that has to be answered concerns the optimal level at which one should try to parallelize the advection sub-model so that the computations are also carried in a vector mode at each processor. Unfortunately, it is very difficult to answer this question. In many cases the answer will depend on the particular computer that is to be used and/or on the particular numerical algorithms that are selected for the advection sub-model. One should probably try several possibilities in order to find the most suitable solution of this problem (for the particular parallel computer with shared memory that is to be used to run the model).

8.3.2. Parallel-vector computations in the chemical sub-model

It has been stated in § 8.2.2 that the loop from Fig. 8.2 can successfully be applied to obtain parallel computations during the execution of the chemical sub-model (because the parameter $NX*NY*NZ$ is normally much greater than the number of processors; this is true even when the model is two-dimensional). However, although there are no problems with the parallelization, one will normally have problems with the vectorization when the processors have vector capabilities. The problems arise during the vectorization of subroutine CHEMI because the loops in these subroutines are rather short (the parameter NSPECIES is, as mentioned before, normally not very large). This means that even if the compiler vectorizes all loops in subroutine CHEMI, the speed of computations will not be very high.

The above discussion shows that the parallelization by using directly the loop from Fig 8.2 can efficiently be used on parallel computers with shared memory whose processors have no vector capabilities. If the processors do have vector capabilities, then the parallelization process must be carried out on a lower level. The choice of the level on which the parallel computations are to be carried out depends both on the particular computer and on the numerical methods used in the chemical sub-model.

If the numerical algorithm is relatively simple (for example, the quasi-steady state algorithm; see Chapter 3), then the computations in the chemical sub-model can be organized as **a sequence of very long loops**. The number of loops in this sequence is NSPECIES, whereas the length of each loop in the sequence is $NX*NY*NZ$ (in fact, the number of loops can often be made less than the parameter NSPECIES by uniting some of the loops). It is relatively easy to get the compiler to run all these loops in the so-called "**vector-concurrent mode**". In fact, on many parallel computers the compiler will automatically produce a code that runs in a vector-concurrent mode when the programming work is properly done. This is probably the simplest way to get vectorized computation on each processor and concurrent computations across the processors of a parallel computer with shared memory. As a rule, it is also efficient to run the chemical sub-model in this mode, but it should be noted that this technique is not always the most efficient one.

When the numerical algorithm used in the model is more advanced (and thus also more complicated), it may not be very easy to obtain high speed computations. However, it should be possible (in principle, at least, because the problem contains so many parallel tasks) to obtain both parallel computations across the processors and vectorized computations at each processor.

8.3.3. Parallel-vector computations in the other sub-models

The other two sub-models, the diffusion sub-model and the deposition sub-model, can be treated similarly to the advection sub-model and the chemical sub-model respectively. The advection and the diffusion sub-models can be treated in a common loop in order either to enlarge the parallel tasks or to obtain more parallel tasks. In addition, the deposition and the chemistry sub-models can be treated in a common loop, again in order either to enlarge the parallel tasks or to obtain more parallel tasks.

The main ideas that can be used in all these options are the same as those described in § 8.2.3. Thus, if the advection module and the chemistry module are well optimized on a given computer, then it will also be easy to optimize the diffusion and the deposition modules.

8.3.4. Parallel-vector computations in three-dimensional models

Efficiency is highly desirable when three-dimensional models are to be run because these are very expensive both in terms of computing time and in terms of storage; see also § 8.2.4. Therefore it is important to achieve a high computation speed during this phase. There is no problem when the processors of the computer that is to be used have no vector capabilities. In such a case the ideas described in § 8.2.4 will in general lead to an efficient code (when these are correctly implemented). Some of the ideas from § 8.2.4 are also easily applicable, with obvious modifications, when the processors of a parallel computer have vector capabilities. However, more efforts may be needed in some parts of the code.

Consider first the calculations along the horizontal grid-lines. The ideas described in § 8.2.4 combined with the ideas described in the previous three paragraphs of this section should, in principle at least, be easily applicable when three-dimensional models are to be run on parallel computers with shared memory whose processors have vector capabilities. The subroutine `ADVECT` must be vectorized when the horizontal advection is treated. The technique described in § 8.3.2 is also applicable in the treatment of the chemical sub-model in a three-dimensional air pollution model. The ideas briefly discussed in § 8.3.3 can also be applied when the diffusion and the deposition parts of a three-dimensional model are to be optimized in order to obtain parallel-vector computations.

Let us consider first the computations that have to be carried out along the vertical grid-lines of the space domain of the air pollution model under consideration by treating the system of ODE's (8.5) for $i = 1, 2, \dots, NX*NY*NSPECIES$. There are many parallel tasks when the loop shown in Fig. 8.10 is used, assuming that a subroutine VERTIC is used to perform the computations for each system of ODE's (8.5). This shows that the use of the loop in Fig. 8.10 will often be a good choice when the processors of the particular parallel computer with shared memory that is to be applied in the runs have **no vector capabilities**.

However, if the processors **do have vector capabilities**, then the situation changes. The main reason for the change of the situation is, as in § 8.3.2, caused by the length of the vectors that are involved in the computations with subroutine VERTIC. These arrays are short, because the number of grid-points along the vertical lines is normally small (the number of horizontal layers in the models is often between 10 and 20). This means that the vectorization process will not be efficient (even when the compiler is telling us that all loops are vectorized).

Therefore the loop given in Fig. 8.10 should not be used on parallel computers whose processors have vector capabilities. It is difficult to replace this loop with another simple generic loop when an efficient vectorization process is required. The vectorization techniques will depend on the numerical methods that are selected. Assume that finite elements are used. Then the small systems (8.5) can be united in larger systems. The number of large systems is NSPECIES; each of them contains $NXNY = NX*NY$ small systems. Assume that this is done and denote $NXNYNZ = NX*NY*NZ$. Then the part of the air pollution model that describes the vertical advection-diffusion process can be rewritten as

$$(8.6) \quad P_j dg_j/dt = H_j g_j + f_j,$$

$$P_j \in \mathbb{R}^{NXNYNZ \times NXNYNZ}, \quad H_j \in \mathbb{R}^{NXNYNZ \times NXNYNZ},$$

$$dg_j/dt \in \mathbb{R}^{NXNYNZ}, \quad g_j \in \mathbb{R}^{NXNYNZ}, \quad f_j \in \mathbb{R}^{NXNYNZ}.$$

The notation used in the description of the systems of ODE's can be explained as follows:

(i) $j \in \{1, 2, \dots, NSPECIES\}$,

(ii) P_j is a symmetric and positive definite block-diagonal matrix that depends only on the finite elements chosen (this matrix contains $NXNY$ diagonal blocks, which are the matrices P_i from § 8.2.4),

(iii) H_j is a matrix that depends on the wind velocities as well as on the diffusion coefficient K_z , which means that matrix H_j is in general time-dependent (moreover, H_j is also a block-diagonal matrix with $NXNY$ diagonal blocks that are the matrices H_i from § 8.2.4),

(vi) $g_j = g_j(t)$ is a vector that contains the values of the concentrations of the j 'th pollutant at all spatial grid-points and at time t ,

(v) $dg_j/dt = dg_j(t)/dt$ is a vector whose components are the time derivatives of the concentrations that are stored in vector $g_j(t)$,

(vi) $f_j = f_j(t)$ is a vector whose components depend on the boundary condition used in the definition of the problem (it will be assumed here that Dirichlet boundary conditions are always specified; i.e. the values of the concentrations are given on the boundaries).

Assume that a subroutine VERT3D that performs one time-step in the integration of one of the ODE systems (8.6) has been developed. Then a parallel computational process can be organized by using the loop shown in Fig. 8.11.

DO 50 I=1,NSPECIES
CALL VERT3D
50 CONTINUE

Figure 8.11
The loop that can be used to perform one time-step
in the integration of one equation of system (8.6)
(NSPECIES is the number of parallel tasks
in this part of the computational work).

Let us try to analyze the effect of the action described above. The small systems (8.5) have been replaced with much larger systems of type (8.6). If $NX=96$, $NY=96$, $NZ=10$, then the number of equations per system is increased from 10 in (8.5) to 92160 in (8.6). This means that the loops of subroutine VERT3D are long, and thus the vectorization of this subroutine will lead to high-speed computations. It should not be very difficult to vectorize subroutine VERT3D. One can, for example, consider the block-diagonal matrices as banded matrices (assuming here that an appropriate ordering is performed) and use the techniques for vectorizing banded matrices described in the previous chapter.

The loop from Fig. 8.11 will lead to efficient computations when the subroutine VERT3D is correctly vectorized and when the number of pollutants NSPECIES either is equal to the number of processors or is a multiple of the number of processors. If this is not the case (if, for example the number of processors is much greater than the number of pollutants), then it would be worthwhile to try to improve the performance by enlarging once more the systems. Instead of considering (8.6) as NSPECIES independent systems of ODE's, one can consider only one very large system of ODE's the order of which is given by the product $NX*NY*NZ*NSPECIES$. If such an action is performed, then very large loops will appear in the solution process, and it should be possible to run these loops efficiently in a vector-concurrent mode (see also § 8.3.2).

It has been mentioned many times that the three-dimensional models lead to very large computational tasks. Consider, for example, the case where $NX=96$, $NY=96$, $NZ=10$ and $NSPECIES=35$ and assume that all equations in (8.6) are united in only one very large system of ODE's. In this case the length of the vectors used in most of the loops is longer than $3.0 \cdot 10^6$. The enormous size of the computational problems associated with three-dimensional air pollution models may impose a strong requirement to exploit special features of the particular computer on which the model is to be run. To illustrate this fact, consider the case where the computer has a cache memory. The references to the cache memory are much faster, but normally only a small amount of data can be kept in cache. Therefore it is efficient in this situation to organize the computations so that **the data locality principle** is satisfied (i.e. one should work as long as possible with the same small amount of data). This explains why the requirements to the programmer are increased very much when three-dimensional models are to be handled.

8.3.5. Some remarks

Many of the parallel machines with shared memory have processors that are able to vectorize the computations. The availability of such a computer ensures a greater degree of flexibility in the efforts to obtain a good performance. However, in this case the task of achieving as efficient performance as possible becomes much more difficult (at least in some of the sub-models of a large air-pollution model). Although it is normally not very difficult to obtain a good performance, it may be very difficult to reach the optimal performance that can be achieved. On the other hand, it is highly desirable (and in some situations absolutely necessary) to obtain the best possible performance, because the LRTAP models are very big, and it is not possible to handle them on a computer (even on a potentially very powerful computer) unless either the highest computational speed is achieved or some physical processes are simplified by unrealistic assumptions. The main ideas that can be used in the efforts to optimize the performance of an air pollution model on a parallel computer with shared memory have been discussed in this section.

A good modulation of the model will also in this situation be very helpful. This has already been emphasized in the end of the previous section. However, the effect of a good modulation of the code becomes even greater when the processors of the parallel machine on which the model is run have vector capabilities, because the complexity of the optimization process is increased. If the processors of the parallel computer that is to be used do not have vector capabilities, then there is only one level of optimization (i.e., one must run as a large part of the code as possible in parallel). The number of levels of optimization is increased to two (both vectorization and parallelization must be achieved) when the processors of the particular parallel computer at our disposal are vector processors.

In some cases it will not be sufficient to achieve vector and parallel computations. This is especially true when the problems are very large. When this is the case, one should also try to exploit some special features of the computer (for example, the cache memory could often be successfully used to improve performance). Of course, the exploitation of the special

features of the computer is dependent, first and foremost, on the computer itself. However, it may also be necessary to select the most appropriate numerical method in order to be able to exploit better the special properties of the particular parallel computer that is to be used.

8.4. Numerical experiments on two ALLIANT computers

Both **the Danish Eulerian Model** and different modules of this model were run on several parallel computers with shared memory. Some of the experiments have been reported in Zlatev [522] or Zlatev et al. [550]. Several experimental runs that were performed on two ALLIANT computers will be presented in this section.

One of the test-problems described in Chapter 4 has been used in all experiments, some results of which will be shown in this section. The test-problem can be used to test the performance of the numerical algorithm during the treatment of the advection, of the chemistry, and of the combination of advection and chemistry. The exact solution of the advection sub-model is known (because a circular wind velocity field with a constant angular velocity is used). The chemistry is simplified; only ten chemical species are involved in the chemical sub-model. This simplified chemical scheme used is discussed in Chapter 3. However, all types of chemical reactions that cause numerical difficulties (fast photochemical reactions as well as fast non-linear reactions) are presented in the simplified chemical scheme. Therefore this is an important test. There are also other reasons for selecting this test-problem: (i) it is rather difficult for the numerical algorithms, (ii) the subroutines that are used in this test are also used in **the Danish Eulerian Model** and (iii) the most time-consuming processes, advection and chemistry, are involved in this test.

The **efficiency** achieved when a parallel computer is used is an important quantity. Let t_1 and t_p be the computing times achieved when one processor and p processors are used. Then the efficiency is measured by

$$(8.7) \quad E_p = 100t_1 / (pt_p) .$$

The efficiency achieved in the runs with more than one processor is given in the tables presented in this section.

8.4.1. Tests on ALLIANT FX/80

The first computer, an ALLIANT FX/80, is a configuration that may have up to eight processors. **Each processor has a vector capability.** Thus, the optimization can be carried out, as mentioned in the previous section, on two levels:

(i) vectorize on each processor,

and

(ii) achieve concurrency across the processors.

The usefulness of the vectorization of the code is demonstrated by the results shown in Table 8.1. It is seen that the speed is increased by a factor of about three (and sometimes by a factor greater than three) when the code is properly vectorized.

From Table 8.1 it is also seen that the results are very good also with regard to the concurrency. The latter issue is important and will be discussed in some more detail in connection with the next table.

The computing times obtained when both the advection and the chemistry are activated are shown in Table 8.1. However, this test-example is very flexible: it allows the user to run the module in four modes: (i) only advection without smoothing, (ii) only advection with smoothing, (iii) both advection and chemistry, and (iv) only chemistry. Runs with all these options were performed. The results are given in Table 8.2. The following conclusions can be drawn by studying the results in Table 8.2 (and also by taking into account many other runs that have been performed with this test-problem or with other test-problems):

(i) Quite different modules are used in the advection part and in the chemistry part of the model. However, in both cases nearly the same level of optimization of the code is achieved; the computing times are reduced by a factor of between four and five. Therefore the optimization level of the code when both the chemistry and the advection are used is also high. This is important: one must optimize all time-consuming parts of a big model.

(ii) The smoothing procedure is only needed when chemistry is to be applied and, moreover, when the chemistry reactions are non-linear. It is seen that the smoothing procedure is not very expensive (but it is unfortunately sequential).

(iii) The fully optimized advection + chemistry code performs more than 15 times faster than the non-optimized (no vectorization, no concurrency) code. It is seen from Table 8.1 that the computing time is reduced from 1071 to 71 seconds.

8.4.2. Tests on ALLIANT FX/2800

The results obtained on the second computer, an ALLIANT FX/2800, with maximum of 14 processors available (but this computer may have up to 28 processors), show that high performance can also be achieved on this computer (see Table 8.3). In fact, the efficiency of the optimized code that is achieved on 8 processors is even greater than the corresponding efficiency achieved on the ALLIANT FX/80; compare the percentages in Table 8.2 and Table 8.3. However, it should be mentioned here that the processors of the ALLIANT FX/2800 have no vector capability.

The results from these two machines are compared for the most complicated case (where both the advection and the chemistry are activated) in Table 8.4. It is seen that the second machine is much faster even when it runs on the same number of processors as the first one.

The additional advantage of the second machine is that the computation speed can be increased by using more processors. An example is given in Table 8.3 where results obtained by using 10 processors are presented. We tested the performance on 14 processors also; see Table 8.4. The computing time is reduced only slightly in the transition from 10 processors to 14 processors (and the efficiency is decreased considerably), because we are exploiting the fact that 10 species are advected and the calculations during the advection are distributed very efficiently to the 10 processors (by using the loop shown in Fig. 8.1). If more species are to be studied (in the big model we have 35 species), then the use of more than 10 processors will be considerably more efficient. The efficiency can also be improved by optimizing on a lower level (see the discussion in § 8.2.1).

Number of processors	Only concurrency	Concurrency + vector
1 processor	1071	309
2 processors	551 (97%)	173 (89%)
4 processors	316 (85%)	104 (74%)
8 processors	208 (64%)	71 (54%)

Table 8.1
Computing times (in seconds) on Alliant FX/80
for advection + chemistry.
The efficiency achieved when more than one
processor is in use is given in parentheses.

Number of processors	Advection without smoothing	Advection with smoothing	Pure chemistry	Advection and chemistry
1 processor	150	178	72	309
2 processors	84 (89%)	101 (88%)	45 (80%)	173 (89%)
4 processors	52 (72%)	62 (72%)	26 (69%)	104 (74%)
8 processors	35 (53%)	42 (53%)	15 (60%)	71 (54%)

Table 8.2
Computing times (in seconds) on Alliant FX/80.
Both concurrency and vectorization are used.
The efficiency achieved when more than one
processor is in use is given in parentheses.

Number of processors	Advection without smoothing	Advection with smoothing	Pure chemistry	Advection and chemistry
1 processor	117	123	47	219
2 processors	62 (94%)	65 (95%)	28 (84%)	120 (91%)
4 processors	36 (81%)	65 (81%)	15 (78%)	69 (79%)
8 processors	24 (61%)	25 (62%)	15 (60%)	54 (61%)
10 processors	17 (68%)	18 (68%)	8 (59%)	33 (66%)

Table 8.3
Computing times (in seconds) on Alliant FX/2800.
The efficiency achieved when more than one
processor is in use is shown in parentheses.

The efficiency achieved with 8 processors of the ALLIANT FX/2800 computer is lower than that achieved with 10 processors (see Table 8.4). This is not a surprise because the chemical scheme involves 10 species and, furthermore, because the loop given on Fig. 8.1 is used. This means that the balance load is nearly perfect when 10 processors are used: each processor receives one task and the tasks are practically equally large. The situation changes when 8 processors are used. In this situations the parallel computations during the treatment of the advection sub-models are performed in two sweeps. During the first sweep, each of the 8 processors receives one task, and thus the efficiency during the first sweep is nearly perfect. There are only two tasks for the second sweep, which means that 6

processors are idle during the second sweep. This explains why the efficiency of the computational process is decreased. However, note that the efficiency is not decreased too much because there are many parallel tasks during the chemical part (thus, the efficiency of the computer during the treatment of the chemical model is high when both 8 and 10 processors are used).

Number of processors	ALLIANT FX/80	ALLIANT FX/2800
1 processor	309	219
2 processors	173 (89%)	120 (91%)
4 processors	104 (74%)	69 (79%)
8 processors	71 (54%)	45 (61%)
10 processors	-	33 (66%)
14 processors	-	49 (49%)

Table 8.4
Computing times (in seconds) on Alliant FX/80 and FX/2800 model 200PMs for
advection + chemistry. The best compiler options were used.
The efficiency achieved when more than one
processor is in use is shown in parentheses.

8.5. Concluding remarks and references concerning efficient runs on parallel machines with shared memory

The optimization of any large air pollution code for runs on parallel machines with shared memory should in principle be possible, because there is a lot of parallelism in such a code, especially when some splitting procedure has been applied in the discretization of the air pollution model that is coded. However, the optimization may be a very tough task both because the air pollution codes are normally very large and because the codes operate with huge sets of data. Therefore, one should begin by finding and optimizing the most time-consuming parts of the computational work. It may be necessary to repeat several times this search for the most time-consuming parts of the computational work (followed by an attempt to optimize them). This procedure is described in **Section 8.1**.

There are two major classes of parallel computers with shared memory. The processors of the first class have no vector capabilities, while the processors of the second class have vector

capabilities. The main ideas that can be applied during the optimization of air pollution codes on the first class of parallel computers with shared memory are discussed in **Section 8.2**.

The processors of the second major class of parallel computers with shared memory have vector capabilities. This allows the developers to carry out optimization on two levels. Assume that an air pollution code has to be optimized for runs on parallel computers with shared memory whose processors have vector capabilities. The main ideas that could be used during the optimization of both the first level and the second level are described in **Section 8.3**. Some more details about running big air pollution codes on parallel computers with shared memory can be found in Zlatev [522], and Zlatev et al. [540,550].

Some test-problems, which are modules containing different subroutines of **the Danish Eulerian Model** (see Harrison et al. [228], or Zlatev et al. [525,527,528,538,539]), have been run on both classes of parallel computers with shared memory. The test-problems contain the most time-consuming parts of **the Danish Eulerian Model**; the advection part and the chemical part (the chemical part being, however, slightly simplified). Some numerical results are presented and discussed in **Section 8.4**.

Many of the problems that are to be solved in parallel when a large air pollution code is to be optimized are rather standard operations (as, for example, basic linear algebra operations and fast Fourier transforms). Good texts concerning the concurrent runs of such standard operations are [177]-[179], [206], [339], [562].

CHAPTER 9

RUNNING MODELS ON MASSIVELY PARALLEL COMPUTERS

"Parallel computation actually reflects the concurrent character of many real-time applications"

David J. Evans: *"Parallel Computing - A New Era"*, In: **"ADVANCES IN PARALLEL COMPUTING"** (David J. Evans, ed.), Vol. 1, pp.1-5. JAI Press, London, 1990.

Massively parallel computers are gaining more and more terrain in the field of large-scale scientific computations, a development that is expected to continue in the future. Therefore it may be useful to test the performance of some large application codes on such computers. Of course, large air pollution models must also perform efficiently on massively parallel computers, especially if such computers are available at the particular site where the code is to be run.

Experiments with several modules from the LRTAP model developed at the National Environmental Research Institute (NERI), the Danish Eulerian Model, were run on a massively parallel computer, a CM-200¹. Some information about this massively parallel computer is given in [290,323]; see also the references listed in [290,323]. Numerical results obtained during these experiments will be discussed in this chapter. All experiments were carried out at the Danish Computing Centre for Research and Education (UNI-C).

It must be emphasized that many if not all of the ideas, which were used during the optimization of the modules of **the Danish Eulerian Model** on the CM-200 computer, can also be applied when the modules are run on other massively parallel computers. In fact, the same modules have also been run on several other well-known massively parallel computers such as CM-5, MASPAP and KSR (this last computer has some very special properties).

The optimization of any large application code for efficient runs on massively parallel computers is in general a very difficult task. There are two major difficulties.

The first of these two difficulties is the requirement to program the code in a special version of the FORTRAN language. This means that programs that have been run on other computers must be rewritten in order to run them on most of the massively parallel computers; as, for example, on CM-200. If the code is very long (and this is in general the case when large application codes are run on a massively parallel computer), then this may be a hard and embarrassing task. The problems connected with rewriting the code have been recognized, and several software developers of different computer companies have prepared some tools for automatic transformation of the programs and subroutines so that the code becomes acceptable for the compiler of the massively parallel computer that is to be used in the runs. Moreover, many massively parallel computers that have been developed during the last two-three years accept codes written in a standard FORTRAN language.

¹ **CM** is an abbreviation of **the Connection Machine**, which is one of the well-known classes of massively parallel computers. There are different CM computers; CM-200 and CM-5 are well known.

The second difficulty is that communication between the processors of massively parallel computers (there are sometimes many thousands processors) is expensive. Therefore the number of communications must be reduced as much as possible. Moreover, the communications must be limited, if this is possible, only to neighbouring processors; the communications between neighbouring processors are the cheapest communications. It should be noted, however, that limiting the communications to those between neighbouring processors will as a rule lead to selecting less accurate numerical methods. This is the main source of difficulties: the requirement for communications only between neighbouring processors and the accuracy requirement work normally in opposite directions, and a reasonable compromise has to be found. The conclusion is that the work should be done carefully. The reduction of the total number of communications as well as avoiding communications between non-neighbouring processors can be achieved only when the following three conditions are satisfied:

(i) the numerical methods are carefully selected,

(ii) the data are correctly distributed between the appropriate processors

and

(iii) the computations are organized in a proper way (so that the communication costs are minimized).

9.1. Basic information on the Connection Machine

All experiments that will be discussed in this chapter were performed on a CM-200 computer. The CM-200 is a massively parallel computer that contains up to 65536 bit serial processors that work asynchronously. The particular computer that has been used in the experiments, the computer at the Danish Computing Centre for Research and Education (UNI-C), contains 8196 bit serial processors. Each processor has up to 1 Mbit of local memory and is able to access other processors through a general purpose high-speed communication router, which utilizes a hypercube communication network of an appropriate dimension. Physically, 16 bit serial processors reside on a single chip. Each pair of chips (or, in other words, each group of 32 processors) shares either a 32 or 64 bit floating point unit. Therefore one can consider a CM-200 with 8196 bit serial processors as a machine configured by $8196/32=256$ computational units that are able to work concurrently and to communicate during the computational process. This means that, in the following part of this chapter, the word "**processor**" will be used to refer not to a single bit processor but to a computational unit consisting of a group of 32 bit processors. This is an appropriate agreement. The term "**computational unit**" will be used only when it is necessary to emphasize strongly the fact that, according to the above agreement, "a processor" is a group of 32 bit processors.

The architecture of the CM-200 is shown in Fig. 9.1. It should be mentioned here that the architecture of many of the other massively parallel computers can be illustrated by a figure that is quite similar to that shown in Fig. 9.1.

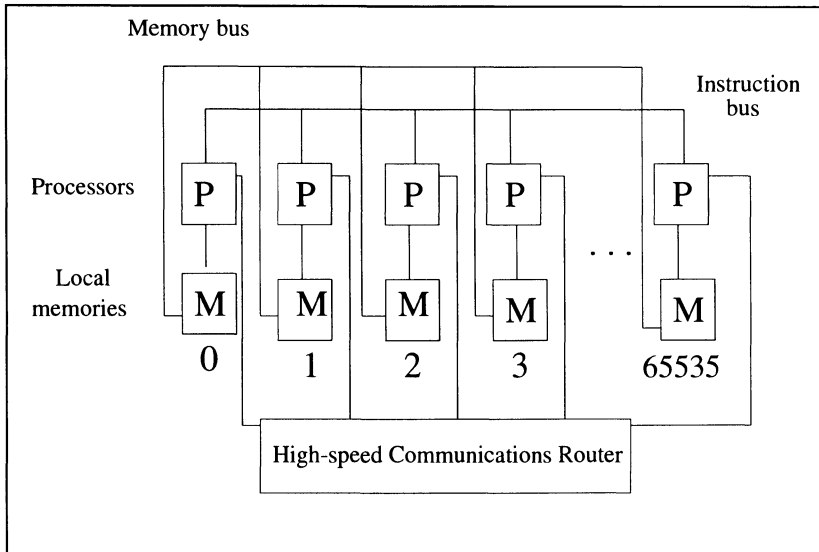


Figure 9.1
The configuration of the CM-200 computer.
Many other massively parallel computers
are configured in a quite similar manner
(i.e. this configuration is typical
for all massively parallel computers).

The theoretical peak performance² of the CM-200 is rather high. In the case when this massively parallel computer is configured with 8196 bit serial processors and when double precision is used, the peak performance is about 2.56 GFLOPS³; see, for example, Levit [290]. However, the speed of computations that is usually achieved when large application packages are run on the CM-200 (and also on all other massively parallel computers that

² *Peak performance of a computer: the maximal computation speed which can be achieved under ideal conditions. The computation speeds on parallel computers (both with shared memory and with distributed memory), which can actually be achieved when large application codes are run, are usually much less than the peak performance.*

³ **GFLOPS:** billion floating point operations, additions or multiplications, per second.

are available at present) is normally much lower. The speed of computations on any massively parallel computer is degraded by:

- (i) communication costs (when information must be sent from one of its processors to other processors),
- (ii) reduction costs (when, for example, norms of vectors or matrices are to be calculated),
- (iii) the fact that different processors have to carry out different calculations (when maskings or the "where" statements are used),
- (iv) some other reasons.

This information is sufficient for our purpose; which is, first and foremost, to explain the numerical results that will be given in the next sections. More details on CM-200 can be found in Levit [290] and in the references given there.

9.2. Mapping large arrays into the memory of a massively parallel computer

Each processor of the Connection Machine has its own local memory (see the previous section and Fig. 9.1). Therefore the perfect situation is the case where

- (i) each computational unit can perform some part of the computations independently from the other computational units

and

- (ii) each computational unit can store in its local memory all data that it will need during the part of computations under consideration.

It is important to emphasize here that both (i) and (ii) must be satisfied in order to obtain high speed computations. It is as a rule not sufficient to have a lot of computations that can be performed in parallel. Normally it is even more important, and in some situations this is the most crucial issue, to have all the data that are needed to perform the computations by a given computational unit of the Connection Machine in the local memory unit of this computational unit.

The processors must use data from their local memories as long as possible. This implies that the arrays where the data are stored (input data, intermediate results and output data) must be shaped so that the data is properly mapped into the local memories of the processors that will use them. Sometimes there is an obvious and natural mapping. Assume, for example,

that 1024 processors are available. Consider the advection part of the two-dimensional model discretized on a (32×32) grid. Then one can think that the processors are connected as (32×32) net. If the data are stored in two-dimensional (32×32) arrays, then it is clear that everything will be mapped perfectly. Unfortunately, the situation is not always so simple, and great efforts are sometimes needed in order to obtain a reasonably good mapping.

In some situations it is appropriate to reshape the arrays in the different parts of the computational process in order to be able to obtain a good mapping for every part of the work. Assume that a three-dimensional air pollution models are to be handled on the Connection Machine. Assume also that the model is split so that, at each time-step, first the computations on the horizontal grid-planes are carried out, and after that the computations along the vertical grid-lines are performed (see more details in Section 3.1). Then reshaping of the arrays may be an appropriate action between the first and the second part of the computations (and such an action has to be carried out at every time-step). The necessity of reshaping the arrays during the computational process is taken into account during the development of software for both the Connection Machine and the other massively parallel machines. On each massively parallel computer there are at present special tools for reshaping the arrays, and these tools are being continuously improved.

9.3. Need to reorganize some computations on a massively parallel computer

Let us assume that some large massively parallel computer is to be used to run a big application code. Then one is interested to carry out as many computations as possible in parallel and to use data from the local memories of the processors as long as possible. These two purposes can be achieved only if the arrays, where the data needed for the computations are stored, are properly mapped onto the net of processors. This is a rather strong requirement (or, at least, could be a strong requirement in some situations). However, in many cases it is not sufficient to map correctly the data from the arrays onto the net of processors of a massively parallel computer. It might also be necessary to re-organize in a new manner some parts the computational process in comparison with the organization of the computations for the old-fashioned sequential machines, and even in comparison with the organization of the computations on the vector machines and/or the parallel machines with shared memory. This could be a difficult task. However, the efficiency of the computational process may depend in a very crucial way on the organization of the computations. An example is given below to demonstrate the usefulness of the reorganization of the computations.

Consider the advection part of a two-dimensional air pollution model involving q species. Assume that the model is discretized on a (32×32) space grid. Assume also that a vector machine is to be used (for example, a CRAY computer or an AMDAHL computer). Then it is worthwhile to store the data (input data, intermediate results and output data) in very long arrays (see Chapter 7). Therefore when vector machines are to be used, it is appropriate to store the data in one-dimensional arrays of length $(32 \times 32 \times q)$ and to design

the computations so that one-dimensional arrays of this length are used. In this way the advection process can also be carried out simultaneously for all q pollutants. If massively parallel machines are to be used, then it is appropriate (assuming here that 1024 processors are available) to store the data in two-dimensional arrays, the length of each dimensions being 32, and to carry out the computations by using such arrays.

This means that, when massively parallel machines are used, it is more efficient to perform the advection process sequentially for each pollutant. If this operation is performed correctly, then the mapping of the arrays onto the set of processors is perfect, and the processors can utilize the local data during the advection step, which consists of many arithmetic operations. However, it is obvious that this requires some re-organization of the computations. Computations involving very long single loops are to be used when the code is run on a vector machine. Computations that consists of triple loops are more efficient on a massively parallel computer. The outer loop runs over the pollutants and is performed sequentially. The double loops run over the grid points and are performed concurrently when the arrays are correctly mapped onto the net of processor of the massively parallel computer used.

9.4. Need to restructure the modules used in the models

The two major requirements that were stated in the beginning of Section 9.2 (to carry out as many operations as possible in parallel and to use local data as long as possible) can often be satisfied by a simple reorganization of the calculations together with appropriate reshaping of the arrays used in the calculations. However, sometimes the efficiency can be increased much more if the modules of the model are restructured by selecting and implementing in them special numerical algorithms that are developed and designed first and foremost for massively parallel machines (and even for the particular massively parallel computer that is to be used).

The most time-consuming operations in the case where massively parallel machines are used in the solution of very large problems arising in science and engineering are:

(1) the communications between processors (i.e., sending data from one processor to some of the other processors),

(2) the reduction operations (the operations where all or many of the components of an array are used to calculate a simple variable; calculations of norms of vectors and matrices are typical examples for reduction operations),

(3) different calculations with different parts of arrays (using different kinds of maskings and using the "**where**" statements in loops are typical examples for this kind of operations).

Therefore special numerical algorithms, where one tries to avoid the use of the above operations or to minimize the numbers of communications, reductions and statements like "**where**", are to be selected and/or developed for large application codes when these are to be run on massively parallel computers.

It should be mentioned here that the situation is gradually improving. More and more numerical algorithms that perform efficiently standard arithmetic operations (as, for example, basic linear algebra operations) on massively parallel computers are already available on different architectures. However, this is often not sufficient. If the particular application code that is to be run on a particular massively parallel computer is very large, then there are still many decisions (in the efforts to achieve the highest possible speed of computation) that are to be taken by the developer of the code. For example, it is not always clear how to store the data in arrays and/or how to reshape these arrays in the different parts of the computational work.

9.5. Some results obtained on the Connection Machine (CM-200)

The Danish Eulerian Model as well as different modules of this model were run on several different high-speed computers (see also the previous two chapters or Zlatev [515] and Zlatev et al. [525,538-540]). Some experimental runs that were performed on a CM-200 computer configured with 8196 bit serial processors (i.e. 256 computational units; see Section 9.1) will be presented in this section.

Some of the test-problems that have been described in Chapter 4 were used in the experiments. The computations connected with these tests are much simpler than the computations in **the Danish Eulerian Model**. However, all difficult parts of the computational process are represented in these test-problems (i.e. the transition from the test-problems to real computations with **the Danish Eulerian Model** will magnify several times the number of simple arithmetic operations, but will not cause some new difficulties). Therefore, **all** conclusions that are valid for the performance of the test-problems on CM-200 will also hold for the global performance when **the Danish Eulerian Model** is run on this computer. Moreover, many of the conclusions are rather general and will also be valid when other massively parallel computers are used.

Consider the general class of test-problems that has been defined in Chapter 4 by the system of PDE's (4.11). Assume that the third type of tasks (advection + chemistry) within this class of test-problems is to be treated. Then the main parts of the computational work during the numerical treatment of (4.11) (see again Chapter 4) are:

- (i) space discretization of the advection terms by chapeau finite elements (systems of linear algebraic equations with tri-diagonal matrices are to be solved in this part),

(ii) time discretization of the advection sub-model (explicit integration methods for solving systems of ODE's based on predictor-corrector methods with several different correctors are applied in this part),

(iii) smoothing the concentrations before the application of the chemical reactions in order to avoid computations with negative concentrations (negative concentrations will normally cause disaster in the treatment of the chemical part),

(iv) treatment of the chemical sub-model (it should be noted here that the simplified chemical scheme with 10 species described in Chapter 2 has been used in the experiments on the Connection Machine).

Let us start the discussion with **the first part** of the computational work. Communications are needed in this part. The simple chapeau finite elements were chosen in order to reduce the number of communications (i.e. linear one-dimensional finite elements have been applied; see Chapter 3). When this choice is made, each processor is communicating only with two neighbours. Thus, the communication cost is nearly minimal. Nevertheless, the speed of computations during this part is considerably lower than that for the second part (see below) when a 32x32 grid is in use. Refining the grid results in improving the performance. Experiments with different grids were carried out in order to confirm this statement; see Table 9.1 and Fig. 9.2

Reductions are also needed during the first part; because the systems of linear algebraic equations (that arise there) are solved by iterative methods and vector norms are needed in the stopping criteria. The main effects of the reductions are very similar to those that are caused by the communications:

(a) the reductions degrade considerably the speed of computations

and

(b) the relative cost of the reductions is reduced when the grid is refined which leads to an increase of the speed of computations; this is illustrated in Table 9.1.

Consider now **the second part** of the computational process described above. No communications and no reductions are needed because simple explicit integration methods for solving ODE's are used in this part. Moreover, the loops are very simple; the same kind of arithmetic operations are used for all components of the vectors that are involved in this part (i.e., no masking is needed and no constructions with `where` statements are used). Therefore the performance of the CM-200 computer is very high when the second part is executed; this is seen in Table 9.1.

Part of the work	G R I D - S I Z E				
	32x32	64x64	128x128	256x256	512x512
Part 1	64	169	320	480	580
Part 2	332	685	910	961	850
Part 3	13	47	136	248	312
Part 4	352	510	560	577	579

Table 9.1

Speed of computations measured in MFLOPS.

Part 1 - space discretization of the advection terms.

Part 2 - time integration of the advection sub-model.

Part 3 - smoothing to avoid negative concentrations.

Part 4 - treatment of the chemical sub-model.

For all grids the number of time-steps is 3456.

The third part of the computational work is not very expensive in comparison with the other parts. It is included here only because it is crucial for the numerical treatment of the chemical sub-model. The concentrations that are to be used as input in the chemical sub-model must be non-negative. However, the advection sub-model produces negative concentrations in sub-domains where the exact concentrations are zeros (or very small numbers). Therefore the concentrations in such sub-domains (when such sub-domains exist) must be smoothed before the beginning of the treatment of the chemical sub-model. It becomes clear from this explanation that smoothing is to be performed only in some parts of the space domain of the model (and these parts must be determined during the call of the subroutine). Therefore, the subroutine that performs smoothing contains all kinds of calculations that delay the speed of computations: communications, reductions, masking, and constructions of loops with the `where` statement in them. This means that one should expect the speed of computations to be very low when this subroutine is executed on CM-200. The results in Table 9.1 show that the speed of computations during the smoothing part are considerably lower than the speed of computations during the other three parts. Nevertheless, it must also be emphasized that the speed of computations becomes greater when the grid is refined also when smoothing is carried out. It must be mentioned here that the smoothing subroutine does not perform very well on other high-speed computers either.

The fourth part is easy for CM-200. There are no communications and no reductions in this part. However, the code contains some loops that are constructed by using `where` statements. The presence of these statements leads to a delay of the computations and, therefore, the speed of computations in this part is not so high as the speed of computations in the second part (the time-integration of the advection sub-model); see Table 9.1.

Some numerical results about the speed of computations that could be achieved in the different parts of the code are shown in Table 9.1. It is assumed, as stated above, that the third type of tasks (advection + chemistry) is treated for a class of test-problems (4.11) that has been introduced in Chapter 4. Five different grids, (32x32), (64x64), (128x128), (256x256) and (512x512), were used in the experiments. It is seen from Table 9.1 that the performance of the Connection Machine is nearly always improved when the grid is refined. The case where the finest grid, (512x512), is used is sometimes an exception. The reason for such a phenomenon is probably the fact that there is not sufficient memory to store all the data that will be needed by a processor in its local memory.

One can consider the total time that is spent on CM-200 (it will be called the CM total time) as a sum of three major components:

- (i) the CM CPU time,
- (ii) the CM communication time,
- (iii) the CM reduction time.

In the rest of this chapter CM will be omitted when there is no danger of misunderstanding. This means that total time, CPU time, communication time and reduction time will often be used instead of CM total time, CM CPU time, CM communication time and CM reduction time.

It is interesting to investigate the relations between these times. An experiment has been performed to get information about the behaviour of the major components of the computing time when the grid is refined. In this experiment the above problem has been run on five grids: (32x32), (64x64), (96x96), (128x128) and (160x160). The computing times (the total time, the CPU time, the communication time and the reduction time) obtained in this experiment are given in Fig. 9.2. It is seen that the first three computing times (the total time, the CPU time and the communication time) are steadily increasing when the grids are refined, while the fourth computing time (the reduction time) is nearly constant.

The relative parts of the three major components of the total time (the CPU time, the communication time and the reduction time) are shown, as percentages of the total time, in Fig. 9.3. The results in Fig. 9.3 are very illustrative. It is clearly seen that the following three conclusions can be drawn by using these results.

No.	Conclusion
1	The relative part of the CM CPU time grows when the number of grid-points is increased.
2	The relative part of the CM communication time stays nearly constant when the number of grid-points is increased.
3	The relative part of the CM reduction time decreases when the number of grid-points is increased.

It is easy to decide (by using these three conclusions) where to concentrate major efforts in the attempt to improve the performance of the code. It must be emphasized here that the rules given below apply not only to large air pollution models, but also to many other problems that lead to large-scale scientific computations.

If one intends to run the model on a very fine grid (because of the accuracy requirements), then one should not bother too much about the reduction time. The communication time is the more important factor in this situation. However, the most important factor in the case where a very fine grid is to be used is the CPU time, and the major efforts in the attempts to optimize the code must be made to reduce the CPU time.

If the grid is not very fine, then one must attempt to reduce both the CPU time and the communication time.

Finally, if the grid is rather crude, for example (32x32), then one must attempt to reduce all three major components of the computing time; i.e. both the CPU time, the communication time and the reduction time.

The results in Table 9.1 as well as those in Fig. 9.2 and Fig. 9.3 demonstrate the performance that can be achieved on a massively parallel computer. They show that high speed on computations can be achieved for the most time-consuming processes in the numerical treatment of large air pollution models when massively parallel computers are used. However, it is also important to show that sufficiently accurate results can be obtained. This is necessary, because the price that is to be paid for the high performance was (in some situations, at least) the use of special methods with a low communication cost. The test-problem allows us to check easily not only the performance that is achieved during the runs, but also the accuracy of the results. Accuracy checks as those described in Chapter 4 can be used, and were used, to study the reliability of the numerical methods that have been designed for runs on massively parallel machines when the test-problems are run on the Connection Machine.

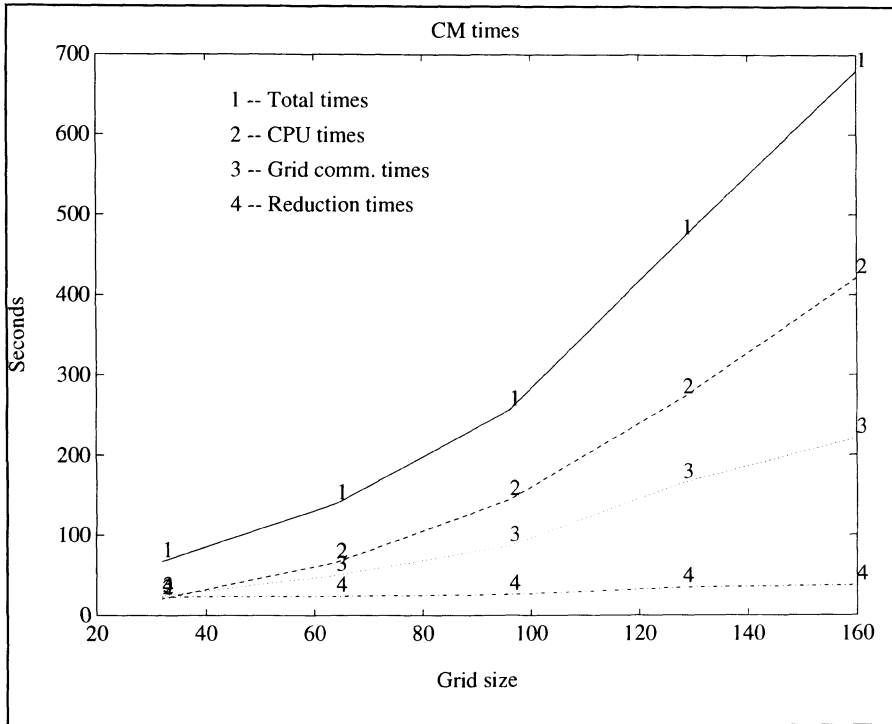


Figure 9.2
Increasing the different CM computing times
with refining the grids.

It should be pointed out that the numerical solution of the system of PDE's, by which the model is described mathematically, is practically useless without a well-developed system of plotting routines that visualize the numerical results. The need of visualization of the numerical results is increased when massively parallel computers, like the Connection Machine, are used because much larger problems can be solved (and are normally solved) when such computers are available. As an illustration of this fact, note that the problems solved in this chapter are much larger than the problems that were treated in the previous chapters. Very often the problems becomes so large that even the static visualization of the results by plots is not sufficient to understand what is really going on. In such a case it is necessary to use animation in order to understand better the trends and the tendencies hidden behind huge sets of output data.

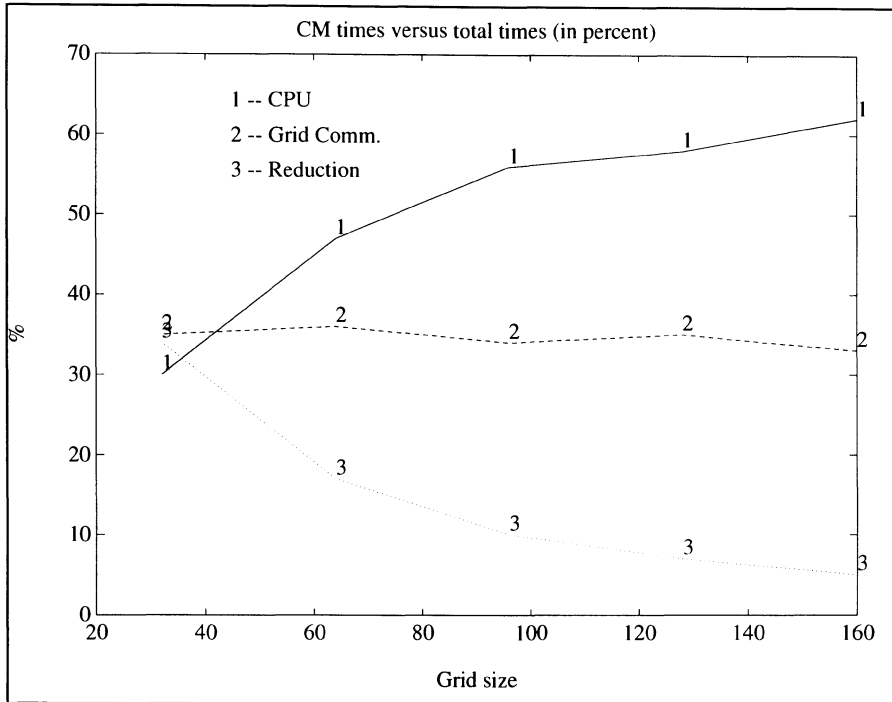


Figure 9.3
Percentages of the most important CM times
(as parts of the total CM time) when the
grids are refined.

9.6. Concluding remarks and references concerning efficient runs of large air pollution models on massively parallel computers

Massively parallel computers are becoming increasingly popular. Therefore it is worthwhile to investigate the possibilities of achieving high performance when large air pollution models are executed on such computers. Some basic information about the configuration of a massively parallel computer is given in **Section 9.1**. Some more details about different massively parallel computers can be found in Levit [290].

Different problems arise when large application codes (not necessarily air pollution codes) are run on massively parallel computers. The main problems are connected

- (i) with the shapes of the arrays used in the code,
 - (ii) with the organization of the computational process (the principles being often different from those used on sequential computers and/or on other types of high-speed computers)
- and
- (iii) with the need to restructure at least some of the modules of the original code.

These difficulties are discussed in **Section 9.2**, **Section 9.3** and **Section 9.4** respectively.

Numerical results from experiments on the CM-200 computer are presented in **Section 9.5**; see also Zlatev and Wasniewski [549]. These results show clearly that massively parallel computers are efficient in the treatment of the most time-consuming modules in large-scale air pollution models. Although the massively parallel machine that has actually been used in our experiments was a CM-200 computer, many of the conclusions made in the previous sections, and first and foremost in **Section 9.5**, are also valid when other massively parallel computers are used. In fact, this was also confirmed by several runs that have been carried out on other massively parallel computers such as CM-5, MASPAP and KSR. The speed of computations, measured in MFLOPS, is quite satisfactory also when other massively parallel computers are used.

All experiments, which were performed on different massively parallel computers (not only these reported in this chapter), indicated also that the results obtained are sufficiently accurate. This is very important, because the accuracy requirements had not the highest priority during the efforts to optimize the code for a massively parallel computer (the speed of computations had the highest priority).

It should not be very difficult to add the other two physical processes, the deposition and the diffusion, to the massively parallel code and thus to run **the Danish Eulerian Model** on CM-200. It is easy to treat numerically the remaining physical processes (diffusion and deposition).

The whole Danish Eulerian Model will soon be run on some massively parallel machine (perhaps on CM-200). However, some practical problems, connected with the need (i) to read huge sets of input data (both meteorological data and emission data) and (ii) to distribute these data among the processors, may appear when such runs are carried out. These data must be read on the front-end machine and moved to the CM-200 after every period of six hour in **the Danish Eulerian Model** (or, in other words, after every group of 24 time-steps). If all meteorological data and the emission data can be stored in the CM-200 data vault, then one should put all input data there and use them repeatedly in the runs. The performance will be improved considerably when this procedure can be used.

CHAPTER 10

NUMERICAL EXPERIMENTS WITH THE DANISH EULERIAN MODEL

"In most cases accurate predictions can only be made after the right mathematical model has been selected".

Garrett Birkhoff: *"Numerical Fluid Dynamics"*. SIAM Rev., **25**(1983), pp. 1-34.

The Danish Eulerian Model has been mentioned many times in the previous chapters of this book. Many of the numerical methods described in the previous chapters are implemented in the modules of this model. **The Danish Eulerian Model** has been run on several high-speed computers. Since both the numerical treatment of **the Danish Eulerian Model** and its performance on several high speed computers have already been discussed in the previous chapters, only some experiments with the model (that have been carried out during the efforts to answer different questions concerning air pollution phenomena in Europe) will be discussed in this chapter.

The exposition of the results will be started by an attempt to locate the position of **the Danish Eulerian Model** in the set of the existing long-range transport of air pollution (LRTAP) models. However, it is necessary to emphasize here that the position of **the Danish Eulerian Model** among the other large air pollution models should not be considered as a fixed position. The situation changes the whole time. **The Danish Eulerian Model** is under a permanent development; either by improving its existing modules or by creating quite new modules that are based on the latest achievements in this area and by adding them to the model. Of course, this is not a big surprise; new and improved procedures are permanently developed and inserted in all other well-known large air pollution models. This process will probably continue a long time in the future.

One of the main reasons for the continuous efforts to improve the performance and the reliability of the large air pollution models is due to the fact that there are still a lot of unresolved problems connected with the different physical mechanisms that are to be taken into account in the models.

Both the fact that many different physical mechanisms have to be tried in order to find the best one for the model under consideration and the fact that the sensitivity of the model results (the concentrations and the depositions calculated by the large air pollution model under consideration) to many different values of the physical parameters has to be carefully investigated increase the requirement for both fast and accurate numerical methods (and, what in many situations will be even more important, the requirement for a correct implementation of the numerical methods in the model). Indeed, it must be reiterated here that

the sensitivity of the output results (concentrations and depositions) that are obtained by a large air pollution model to variations of different physical parameters and procedures could be judged correctly only if it is reasonable to expect that the results obtained by the model are not corrupted by errors due to the application of one or more numerical algorithms that either are not sufficiently accurate or are not correctly implemented in the model.

10.1. The position of the Danish Eulerian Model among the other LRTAP models

"The costs of controls are sufficiently high that society will wish assurance that imposed controls will be effective in reducing air pollution levels. It is thus vital that the overall planning process allow sufficient time and resources for conducting thorough evaluations of model performance".

Ph. M. Roth, S. D. Reynolds, Th. W. Teshe and R. L. Dennis: "A Conceptual Framework for Evaluating the Performance of Grid-based Photochemical Air Quality Models". In: "AIR POLLUTION MODELING AND ITS APPLICATION IX" (H. van Dop and G. Kallos, eds.), pp. 49-56. Plenum Press, New York-London, 1992.

Roughly speaking, there are two big groups of LRTAP models. The first group is the group of the so-called **comprehensive LRTAP models**. Typical representatives of this group are the Canadian model ADOM¹ (Venkatram et al. [459]), the American model RADM² (Chang et al. [66], Stockwell et al. [441]) and the German model EURAD Model³ (Ebel et al. [144]). In these models one attempts to describe in detail all physical phenomena that appear during the long-range transport of air pollutants. This leads to huge computational problems. Therefore, it is not possible, at present at least, to run such models over long time-intervals, but the development of the computers is very fast. Therefore the comprehensive LRTAP models are perhaps the models of the future, and it will be no big surprise if many (or even most) of the LRTAP models will be comprehensive after about 10 years (see, for example, Zlatev et al. [539]).

¹ **ADOM:** Acid Deposition and Oxidation Model.

² **RADM:** Regional Acid Deposition Model.

³ **EURAD Model:** European Reaction Acid Deposition Model.

The second large group of LRTAP models is the group of simple models. The EMEP⁴ model (see Eliassen and Saltbones [154] or Iversen et al. [252-254]) is a typical representative of this group. During the last years, the usual procedure is to incorporate more and more additional modules in the simple models. For example, some more chemical reactions were recently added to the EMEP model (Iversen et al. [252-254]). In this way, the gap between the simple models and the comprehensive models has been reduced during the last years and this process is continuing.

The simple models can be run over long time-intervals. Thus, the seasonal variations of some air pollutants can (in principle, at least) be studied by these models. The comprehensive models can be run, as mentioned above, only on relatively short time-intervals. On the other hand, it is possible to study successfully diurnal variations of some of the pollutants during episodes⁵ when such models are used (because the physical processes are in general described in a more detailed form in the comprehensive models). It is highly desirable, however, to have a model that can be used to study both seasonal variations and diurnal variations of the concentrations involved in the model. Moreover, it is also necessary to include all important air pollutants in the model (because there are many interconnections between different pollutants during the transport; this topic has been discussed in Chapter 1 and Chapter 2, its complexity is illustrated in Fig. 1.1). Therefore, one should try to satisfy at least the three conditions that are described in Table 10.1 in the efforts to improve the performance of a given large air pollution model.

Condition	A C T I O N
1	One should try to add to the model as many as possible of the important air pollutants.
2	One should try to achieve reliable results when diurnal variations are studied.
3	It should still be possible to study seasonal variations with the enhanced model.

Table 10.1
Three important requirements for improving an LRTAP model.

⁴ **EMEP:** European Monitoring and Evaluation Programme. This is a common European air pollution project under the European Commission for Economics at the United Nations. Nearly all European countries are participating in this project. The EMEP models (including here the photooxidant model developed by Simpson [413,414]) are operationally used at the Meteorological Synthesizing Centre - West (MSC-W), which is located at the Norwegian Meteorological Institute in Oslo.

⁵ **Episode:** An episode is a short time-period, normally a couple of days, in which the concentrations of certain air pollutants are very high. It is important to be able to study in detail the variation of the concentrations during an episode.

The third of the conditions stated in Table 10.1 is very often neglected. This may, potentially, result in wrong conclusions. Assume, for example, that a comprehensive air pollution model have extensively been tested on data concerning episodes in August and gives good results for these episodes. If the third condition is neglected (and one does not know whether the model is able to calculate correctly the seasonal variations of the concentrations or not), then the same model may give poor results when, say, episodes that take place in May are studied (because it has been tuned to meteorological, climatological and other conditions that are typical for the late summer). Thus, the requirement to be able to run the model on long time-periods is (or, at least, may be in some situations) essential.

The Danish Eulerian Model was in the very beginning a simple model. It has been initiated as a simple sulphur pollution model (Zlatev [515]). Some modules for calculating nitrogen pollution have been added to the model after 1984. These modules are described in Zlatev [520,522] and Zlatev and Christensen [536]. Gradually, many other new modules have been added to the model (as, for example, a chemical scheme with non-linear chemical reactions, including here also photochemical reactions; see Hov et al. [246]). The development of **the Danish Eulerian Model** continues. The main idea is to continue to keep the upgraded model well-structured, so that it should be possible both to add and to remove in an easy way some modules when such an action is appropriate. In this way the model will probably (sometime in the future) be upgraded to a comprehensive model. However, this is not the most important aim. As pointed out above, it is also both desirable and important to have the possibility of running the model on long time-intervals and, thus, to be able to study seasonal variations of the concentrations.

By upgrading the model in small steps (i.e. by adding and/or by removing one simple module per step), it should be possible, in principle at least, to determine the most economical set of modules that can be used for a particular task. Thus, having the task, one should be able to solve it in an optimal way. To illustrate the usefulness of such an approach, let us mention the case where country-to-country annual sulphur budgets are to be calculated. The optimal way to solve this task is, perhaps, to apply the simple EMEP model (Eliassen and Saltbones [154]).

It will be shown in this chapter that reasonably good results can be achieved in the efforts to satisfy the three important conditions shown in Table 10.1 by adding (to the simple model described in Zlatev [515,517,520,522]) several extra modules; as, for example, modules containing non-linear chemical reactions, modules for treatment some additional meteorological data, modules with a refined spatial grid-discretization, etc.

However, it is also necessary to point out here that the task stated by the conditions in Table 10.1 is indeed very ambitious, and there is no claim that this task has been completely solved in the present version of **the Danish Eulerian Model**. What is claimed here, however, is the following: during the small steps towards the solution of this task, one can improve the results obtained by the model considerably; even in the case where the task of satisfying the conditions in Table 10.1 is not completely solved.

It should also be emphasized that, during the attempts to improve both the performance and the reliability of **the Danish Eulerian Model**, the possibility of running this model on long time-intervals has always been preserved (and the intention is to preserve always this

possibility; also in the future development). Thus, the requirement of preserving this possibility is considered as the most important constraint during the development of the model (i.e. any small step during the development of the model, by which this constraint will be violated, is automatically declared as unacceptable).

10.2. Calculations of the levels of the concentrations and depositions of air pollutants in Europe

"We rely on numerical models to address problems that cannot be solved by empirical means. The processes involved in the production, transport and deposition of atmospheric acids are sufficiently complex that we cannot hope to understand acid deposition source-receptor relationships definitively without resorting to models".

D. A. Hansen, H. M. Barnes, M. Lusi and K. J. Puckett: "A North American Field Study to Evaluate Eulerian Models". In: "AIR POLLUTION MODELING AND ITS APPLICATION VII" (H. van Dop, ed.), pp. 297-306. Plenum Press, New York-London, 1989.

The space domain of **the Danish Eulerian Model** contains the whole of Europe (see Fig 2.1 in Chapter 2). The model has been developed to study the long-range transport of air pollutants in Europe. Therefore the first requirement to the model is to give information about the levels of the concentrations and depositions of the most important air pollutants involved in the model within its space domain (i.e. mainly within Europe). The ability of the model to calculate the levels of the concentrations and depositions of the air pollutants in Europe will be demonstrated in this section by showing results obtained by running the model with meteorological and emission data for 1989. It should be pointed out here that similar runs have been performed with input data (both meteorological data and emission data) for 1985, 1990 and 1991.

The input files used in the model are given in Table 10.2 (the meteorological files needed) and in Table 10.3 (the emission files needed).

The meteorological data, see again Table 10.2, is read in the beginning of every period of six hour (excepting here the mixing heights; information about mixing heights is read in the beginning of every period of twelve hours). Simple linear interpolation is used when necessary.

The emission files that are used in the model are listed in Table 10.3. It should be noted that the emission files that are available at present contain only annual mean values of the emissions. Special interpolation procedures are used in the model to simulate some temporal variations of the emissions; see Zlatev et al. [534,535,538,539].

Meteorological input data	Source
Advection wind	EMEP
Vertical wind	EMEP
Mixing height	EMEP
Average precipitation	EMEP
Cloud cover	EMEP
Temperature in mixing layer	EMEP
Surface temperature	EMEP
Relative humidity	EMEP

Table 10.2
Meteorological files used in the model.

Emission input data	Source
Sulphur emissions	EMEP
Nitrogen (NO _x) emissions	EMEP
Ammonia (NH ₃) emissions	[11]-[13]
Anthropogenic VOC emissions	EMEP
Natural VOC emissions	EMEP

Table 10.3
Emission files used in the model.

Different ways to visualize the output files can be applied. One can use scatter plots, where the concentrations and depositions calculated by **the Danish Eulerian Model** are compared with measurements. One can also plot time-series giving the variations of different quantities at the measurement stations (and compare them with corresponding time-series obtained by using measurements when measurements are available). The lengths of the periods of averaging that could be used in the plots are given in Table 10.4. It should be mentioned here that hourly mean values are only used in connection with the ozone concentrations, because only for ozone concentrations observations on hourly basis have been available until now at the National Environmental Research Institute (Roskilde, Denmark).

Length of periods of averaging
Hourly mean values
Daily mean values
Monthly mean values
Seasonal mean values
Annual mean values

Table 10.4
Time-averages used in the output files.

The different quantities that are available (as output from the Danish Eulerian Model) are given in Table 10.5.

Output quantities
Air concentrations
Concentrations in precipitation
Wet depositions
Dry depositions
Total depositions

Table 10.5
Different quantities available in the output files.

More details about the input data used (the meteorological data and the emission data) can be found in Zlatev et al. [538,539]. There is an essential difference between the results in Zlatev et al. [528,539], and the results in this section. The results presented in Zlatev et al. [538,539], were obtained by using a (32x32) grid to discretize the space domain. Most of the results that will be presented in this section are obtained by using a (96x96) grid. This means that one should expect some gradients of the concentrations and depositions to be more correctly calculated (because a finer grid is used).

The space domain of the model (see Fig. 2.1 in Chapter 2) contains not only Europe, but also parts of Asia, Africa and the Atlantic Ocean. However, in many cases it would be much more informative to present plots where only the results on some sub-domain of the space domain given in Fig. 2.1 are presented. In the cases where this is done, it would be possible to show the grid discretization together with the particular values of the con-

centrations and the depositions found in the grid-squares. The model has often been used to study the concentrations and depositions in Denmark and in the surroundings of Denmark. A part of the space domain that contains Denmark could be chosen in this situation (and this will be done in Section 10.7). However, it should be emphasized that a part of the space domain that contains Denmark is not the only possible sub-domain that can be specified. Any other sub-domain within the space domain of the model can be chosen; the whole space domain can also be chosen (and this is often done when colour plots are drawn). In fact, there is a rather flexible visualization programme in which one will interactively be asked to give

- (i) the pollutant chosen,
- (ii) what is to be plotted (air concentrations, concentrations in precipitations, wet depositions, dry depositions or total depositions),
- (iii) the coordinates of the left lower corner of the desired sub-domain together with its size (the number of grid-squares along the side of the sub-domain that is parallel to the Ox axis).

Then the visualization programme automatically performs everything: plots the grid, draws the map and inserts the concentrations or depositions required. There are many other options in the visualization programme (for example, one can omit the grid, one can draw colour plots or plots in black-white, one can insert the values of the concentrations and depositions in every grid-square or omit them, one can draw niveau-curves, etc.).

Several plots are given below to demonstrate the ability of the programme to visualize the results calculated by the model. The levels of the mean values of the sulphur dioxide concentrations and the nitrogen dioxide concentrations in Europe (for 1989) are given in Fig. 10.1 and Fig. 10.2 respectively. The whole region is drawn in these two figures. The case where only a part of the region is taken into consideration (when plots are drawn) will be illustrated, as mentioned above, in Section 10.7.

10.3. Comparisons with measurements

"The operational evaluation uses network data or long-term averages to determine how well the model performs on such questions as total sulphur and nitrogen deposition".

A. M. Macdonald, C. M. Banic, W. R. Leaitch and K. J. Puckett:
"Evaluation of the Eulerian Acid Deposition and Oxidant Model (ADOM) with Summer 1988 Aircraft Data". Atmospheric Environment, 27A(1992), pp. 1019-1034.

The accuracy of the numerical methods used in **the Danish Eulerian Model** has been checked by using many of the test-problems described in Chapter 4. Local error estimators (as those discussed in Chapter 5) have been used in some modules of the model. Therefore

one should expect the numerical algorithms that are applied in the model to be sufficiently accurate. Nevertheless, one extra check is absolutely necessary: one must also compare the results obtained by the model with measurements taken in stations located in different European countries. There are two main reasons for such an extra check:

(a) the great uncertainties in the input data (which is true both for the meteorological data and for the emission data)

and

(b) the many open questions in the description of the particular physical mechanisms that are to be modelled and in the determination of particular values of the physical parameters that are to be used in the model under consideration.

This extra check (the comparisons of model results, concentrations and depositions, with corresponding measurements) is commonly used in practice (in an attempt to evaluate the reliability of the concentration and deposition fields that are calculated by a large air pollution model).

Comparisons of concentrations and depositions calculated by **the Danish Eulerian Model** with corresponding measurements have been carried out. Some of the results obtained in these comparisons will be presented in this section. Two types of comparisons with measurements will be demonstrated in this chapter:

(i) scatter plots

and

(ii) time variations.

10.3.1. Scatter plots

Scatter plots are traditionally used to compare calculated and measured concentrations and/or depositions. Every station is usually plotted as a dot in the scatter plot. One can number the measurement stations and the numbers of the measurements stations can be drawn on the scatter plots (instead of using dots). In both cases, the location of the measurement station on the plot is determined by two parameters: the measured concentration or deposition at the station under consideration and the calculated by the model concentration or deposition (in the grid-square where the station under consideration is located). Thus, the ideal case is a plot where all dots (or numbers) by which the measurement stations are represented lie on the line through the origin of the coordinate system and such that the angle between it and the Ox axis is 45° . Such a line is drawn on all scatter plots given in this section. It should be stressed, however, that this ideal case will never occur in practice.

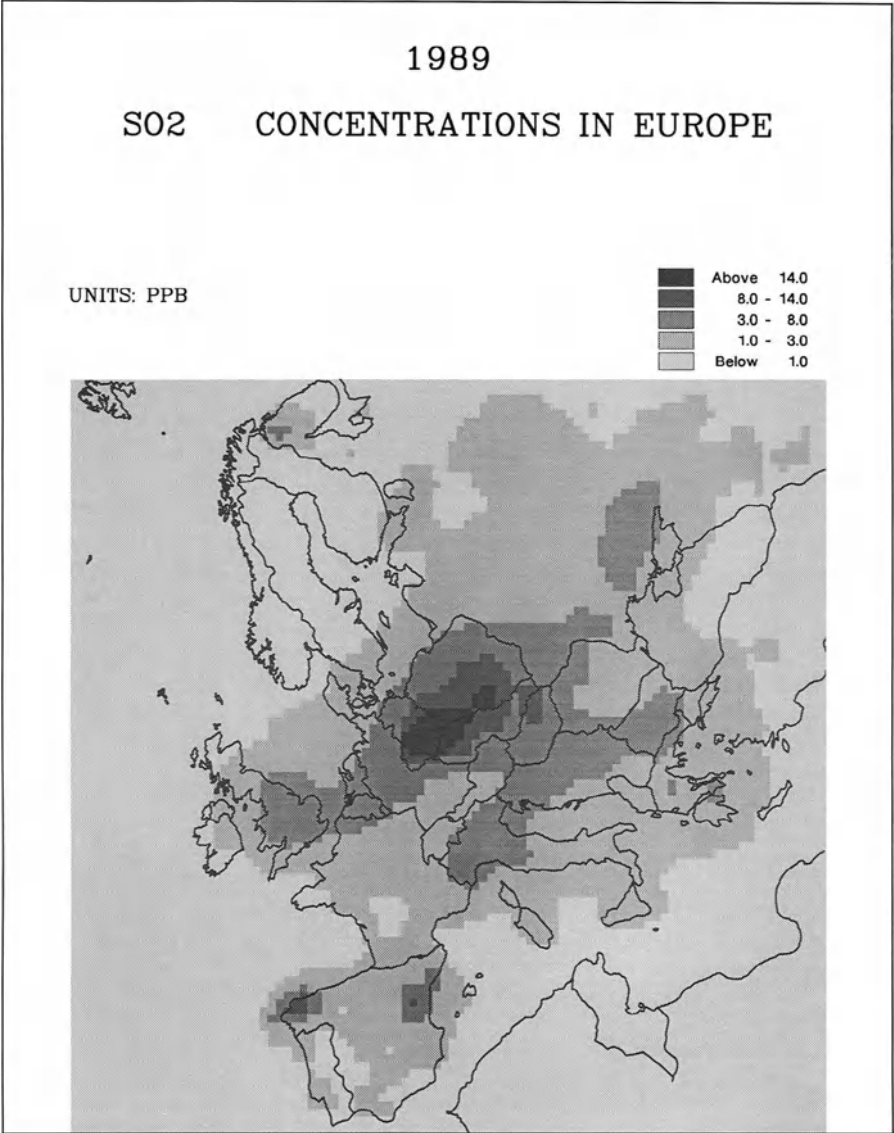


Figure 10.1
The sulphur dioxide concentrations in Europe for 1989.

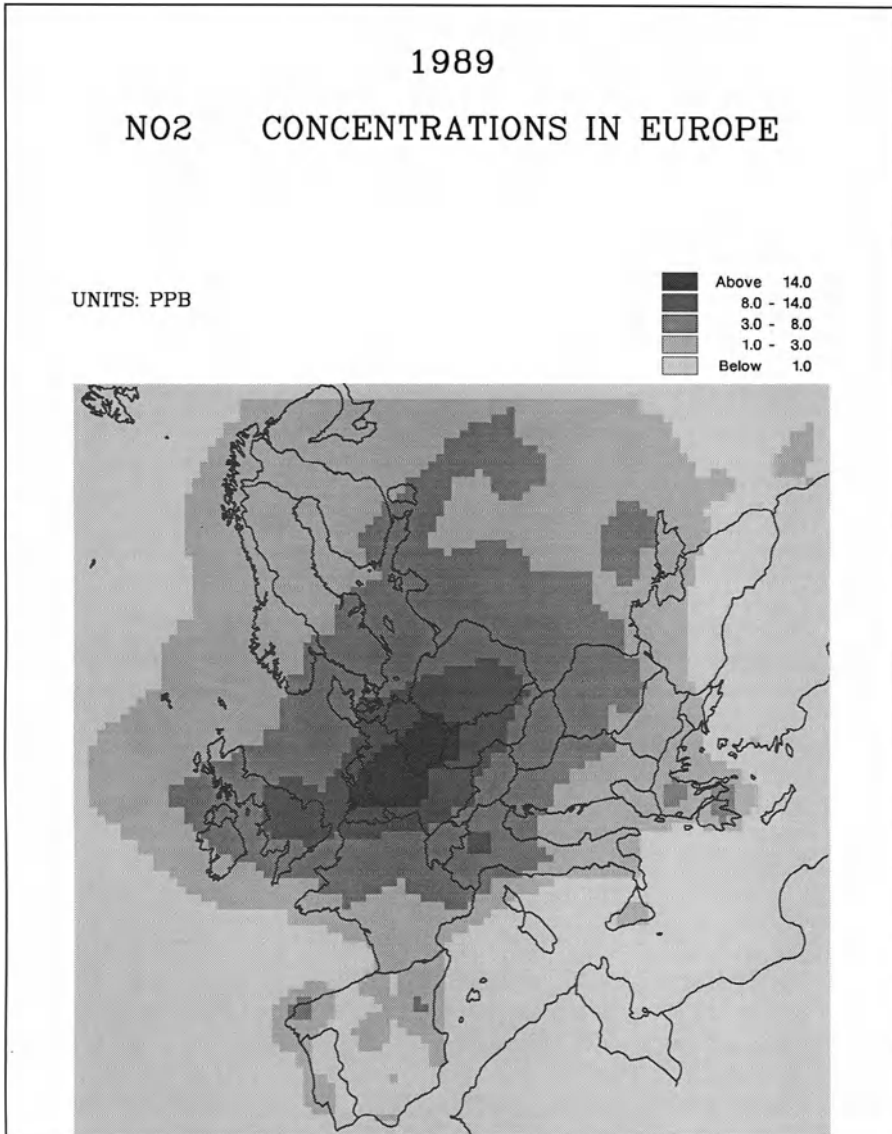


Figure 10.2
The nitrogen dioxide concentrations in Europe for 1989.

It is important to answer the question whether the observations at a given measurement station are representative or not. The problems that are connected with the representativeness of the observations made at a given measurement station (when these are to be compared with results obtained by a given large air pollution model) are very difficult and, in general, unresolved.

In this section, it is assumed that the observations (concentrations or depositions of a given pollutant) measured at a given station are representative if the following three conditions are satisfied:

(i) The measurements must be regular. More precisely, this means that at least 15 observations per month must be available (and this condition must be satisfied in every month of the year of consideration) when air concentrations are studied. The corresponding number for the concentrations in precipitation is five.

(ii) The station should not be located too high (no measurement station which is above 1200 over the sea level is taken into account in the experiments presented in this section).

(iii) The station should not be located close to the boundaries of the space domain (the model results obtained in parts of the space domain that are close to the boundaries are uncertain).

Scatter plots, where the concentrations and depositions calculated by the model have been compared with measurements taken at the EMEP stations that satisfy the above conditions on annual, seasonal and monthly bases, have been drawn (for several different years; therefore it is impossible to present all results here).

Numerical results that are obtained in some of the comparisons for 1989 are given in Fig. 10.3 - Fig. 10.14. The annual means of the sulphur dioxide and nitrogen dioxide concentrations are compared in Fig. 10.3 - Fig. 10.4, respectively. Seasonal means of the same concentrations are compared in Fig. 10.5 - Fig. 10.6 (winter), Fig. 10.7 - Fig. 10.8 (spring), Fig. 10.9 - Fig. 10.10 (summer) and Fig. 10.11 - Fig. 10.12 (autumn). Mean values of the calculated and observed concentrations for September 1989 are compared in Fig. 10.13 (sulphur dioxide) and Fig. 10.14 (nitrogen dioxide).

It should be mentioned that similar plots have been produced and analyzed for the other species for which there are measurements and for the other months of 1989 (as well as for some months in 1990 and 1991).

It is seen that in general the model gives rather reliable results on annual basis. The quality of the comparisons is decreased when the averaging period becomes smaller (i.e. when seasonal and monthly mean values are compared), but in general the results are still quite acceptable.

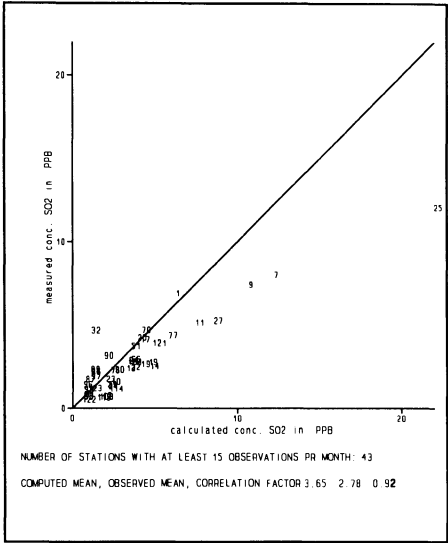


Figure 10.3
Calculated and measured sulphur dioxide concentrations for 1989.

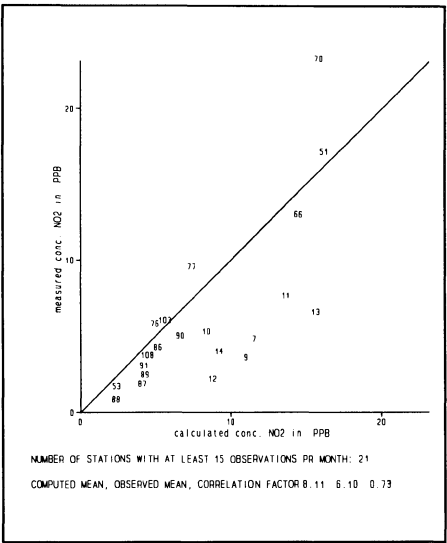


Figure 10.4
Calculated and measured nitrogen dioxide concentrations for 1989.

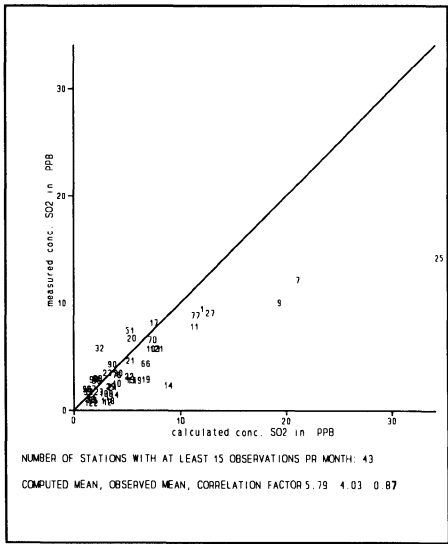


Figure 10.5
Calculated and measured sulphur dioxide concentrations for winter 1989.

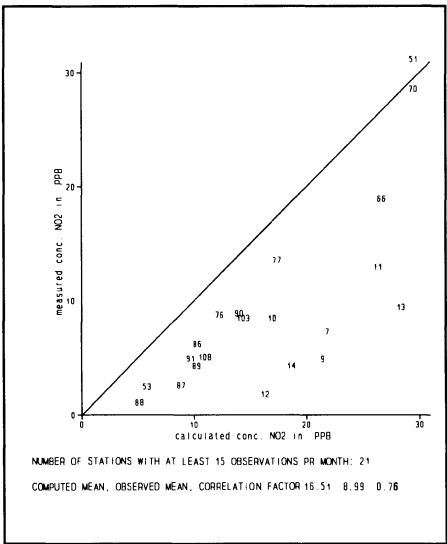


Figure 10.6
Calculated and measured nitrogen dioxide concentrations for winter 1989.

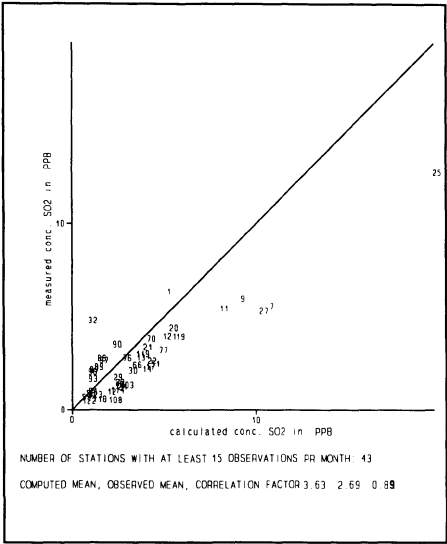


Figure 10.7
Calculated and measured sulphur dioxide concentrations for spring 1989.

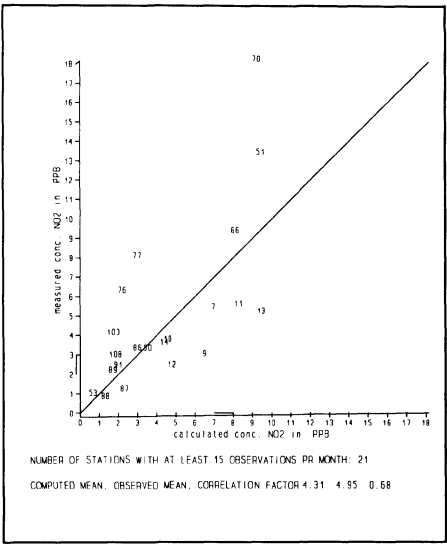


Figure 10.8
Calculated and measured nitrogen dioxide concentrations for spring 1989.

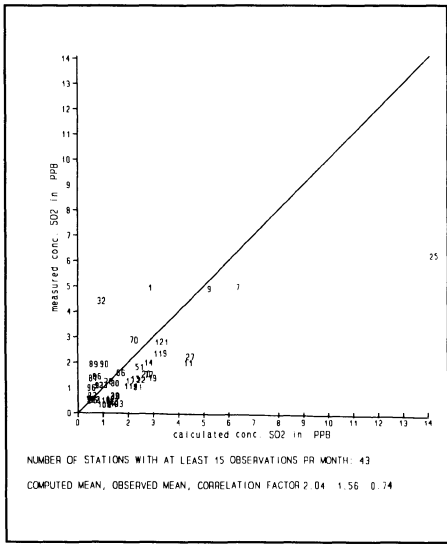


Figure 10.9
Calculated and measured sulphur dioxide concentrations for summer 1989.

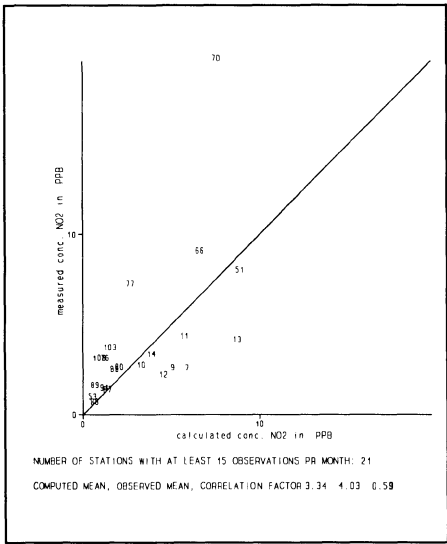


Figure 10.10
Calculated and measured nitrogen dioxide concentrations for summer 1989.

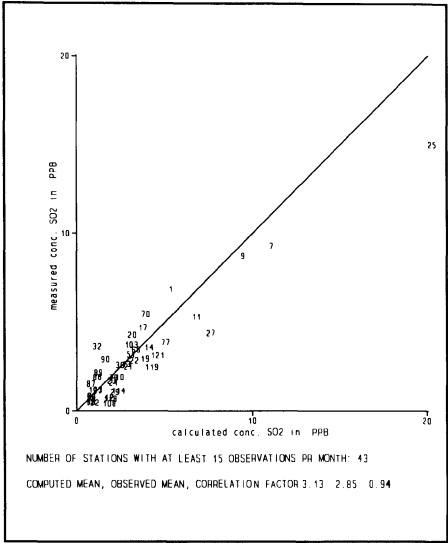


Figure 10.11
Calculated and measured sulphur dioxide concentrations for autumn 1989.

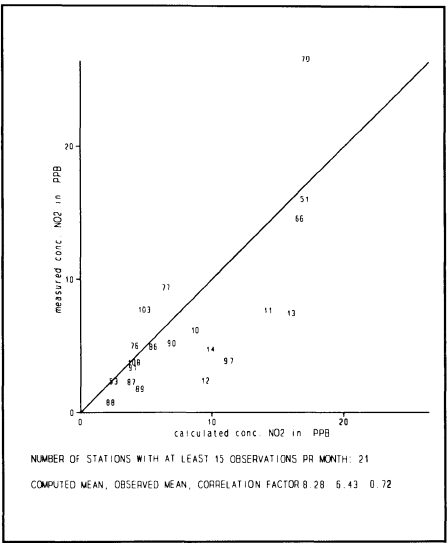


Figure 10.13
Calculated and measured nitrogen dioxide concentrations for autumn 1989.

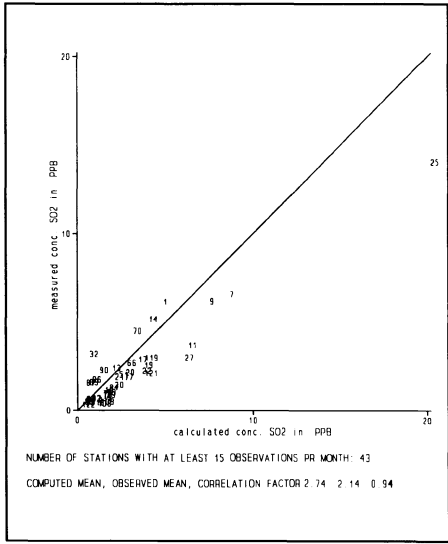


Figure 10.12
Calculated and measured sulphur dioxide concentrations for September 1989.

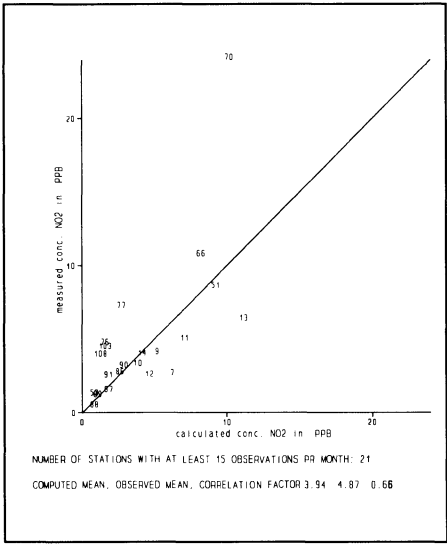


Figure 10.14
Calculated and measured nitrogen dioxide concentrations for September 1989.

The model seems to overestimate slightly the sulphur dioxide concentrations. The overestimation of the nitrogen dioxide concentrations is sometimes more considerable. For example, the model overestimates the nitrogen dioxide concentrations in winter rather significantly. The chemical scheme (the condensed CBM IV scheme) is probably not very suitable for calculations in winter periods.

10.3.2. Time variations

It is also interesting to check whether the model results calculated at a given measurement station follow the corresponding measurements. This is a much more difficult test. One of the major difficulties arises from the fact that the model results are representative means for the grid-square where the measurement station under consideration is located (the area of the grid-square being 2500 km² when the new version of **the Danish Eulerian Model** is used), while the measurements are point-values (taken at the station under consideration). Still, the results indicate that the model is able to calculate concentrations that follow rather well (or, at least, not too bad) the variations of the concentrations at the most of the measurement stations in the EMEP network. It should be emphasized that this is true not only when the mean monthly values are used in the comparisons, but even if daily mean values (calculated and measured) are compared. The improved accuracy obtained during the comparisons of the time variations of calculated and measured concentrations is due to the use of a finer grid in the space discretization.

A careful study of the daily variations of calculated and measured concentrations at many of the EMEP stations indicates that there are short periods (one-two days) in which the calculated by the model sulphur and nitrogen concentrations overestimate considerably the corresponding measurements in some parts of the space domain. In July 1989 there are two such periods (in fact only two days: July 7 and July 23) for the area containing Southern Sweden, Denmark and most of the Germany. Some results for sulphate are given in Fig. 10.15 - Fig. 10.18.

Similar results have also been observed, in July 1989, for sulphur dioxide and nitrogen dioxide, but the maxima for these two pollutants are not so pronounced as the maxima for sulphate.

Let us consider the time series of calculated and measured sulphate concentrations that are given in Fig. 10.18. It is not very clear why the model gives quite acceptable results for 29 days in July, while it overestimates rather considerably the concentrations, in this example the sulphate concentrations, in two particular days (July 7 and July 23). An attempt to study the meteorological data files for July 1989 has been carried out. The only essential issue that has been observed until now is the fact that the horizontal wind velocities in the part of the space domain under consideration (Southern Sweden, Denmark and Germany) are very low. This may be the reason for the overestimation of the concentrations by the model. However, more experiments are needed in order to confirm (or reject) such a conjecture. The wind field for July 7 1989 is given in Fig. 10.19.

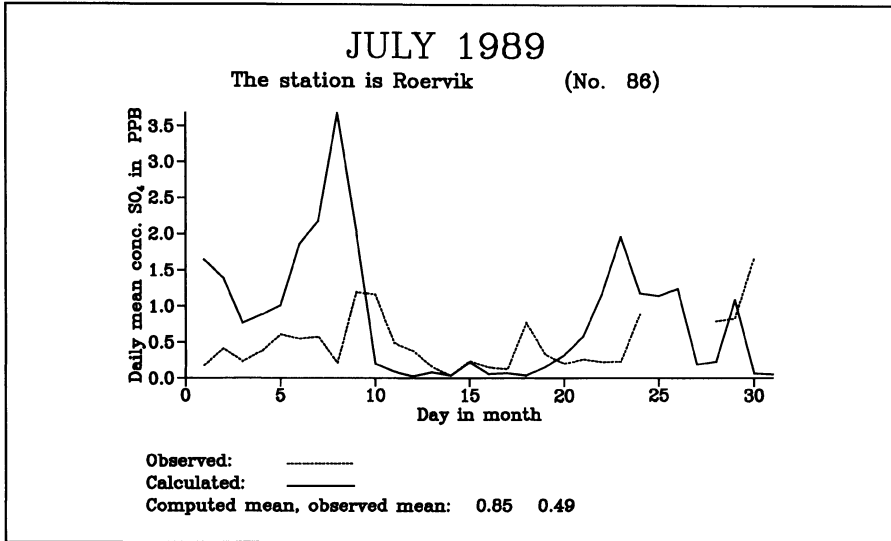


Figure 10.15

Time-variation of daily means of calculated and measured sulphate concentrations.

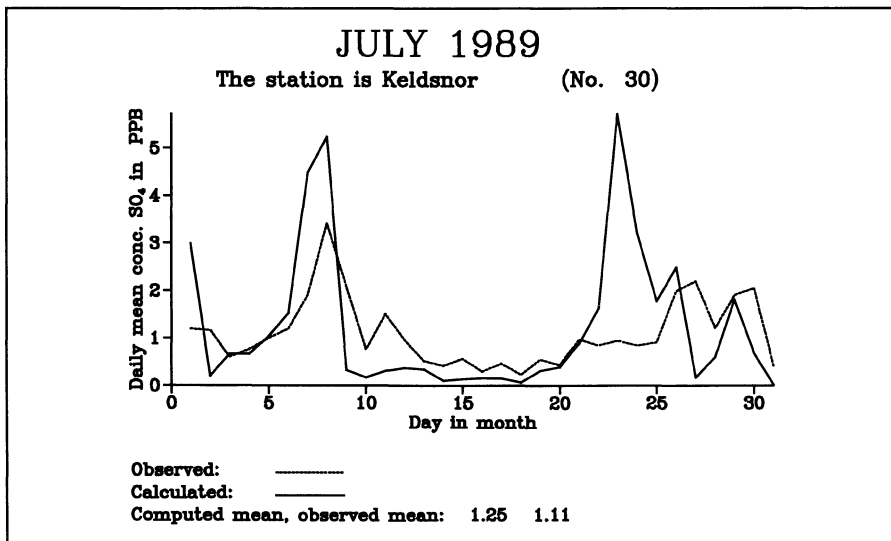


Figure 10.16

Time-variation of daily means of calculated and measured sulphate concentrations.

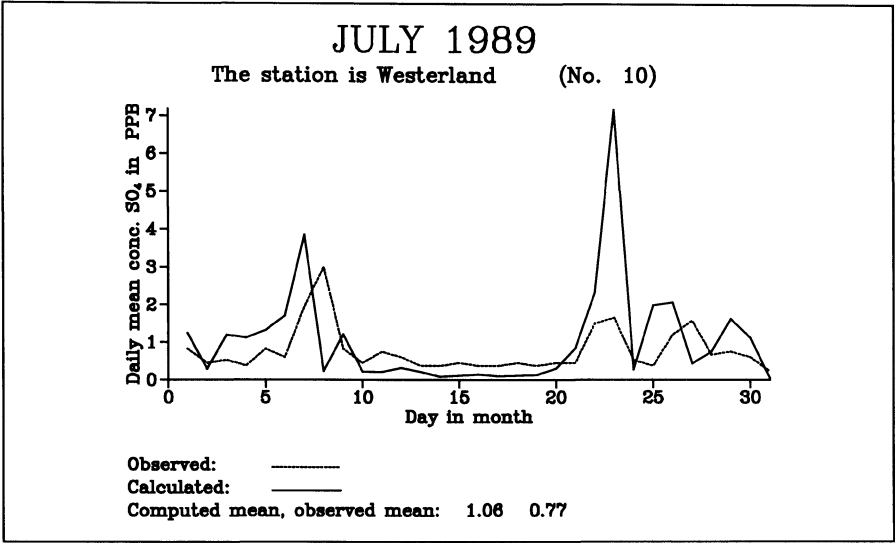


Figure 10.17
Time-variation of daily means of calculated and measured sulphate concentrations.

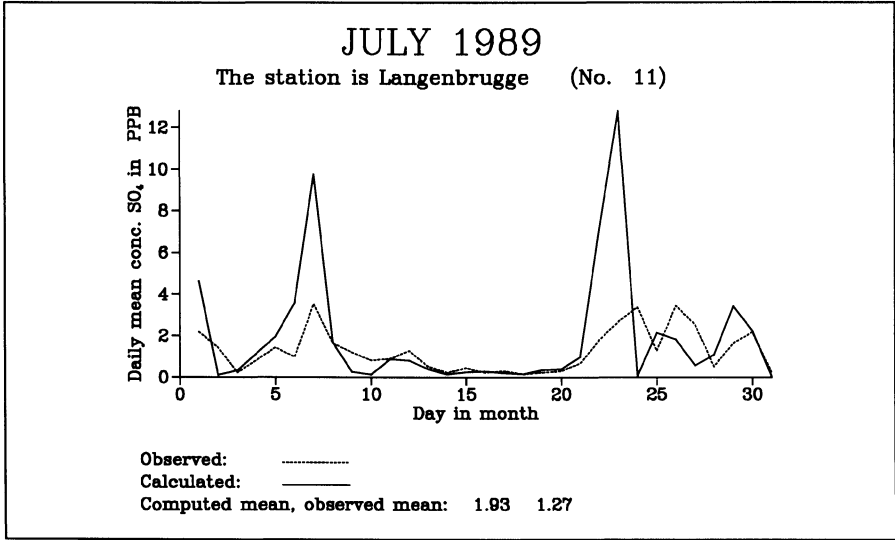


Figure 10.18
Time-variation of daily means of calculated and measured sulphate concentrations.

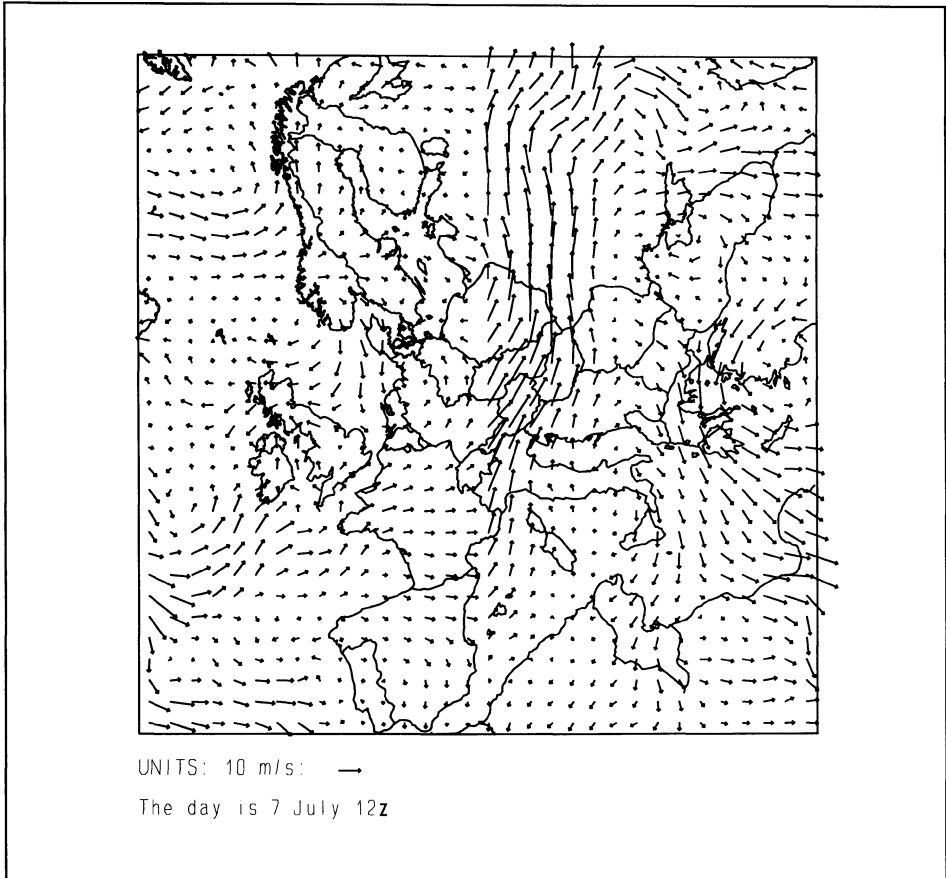


Figure 10.19
One of the wind fields used in the model.

10.4. Studying the influence of sources in one region on the air pollution in the surrounding regions

"Because of the regional nature of the problem the assessment of the source receptor relationships required for initiation of control measures is difficult and cannot be derived from observational data".

R. Stern, B. Scherer and J. Pankrath: *"Application of a Regional Model for the Transport and Deposition of Acidifying Pollutants to Central Europe"*. In: **"AIR POLLUTION MODELING AND ITS APPLICATION VI"** (H. van Dop, ed.), pp. 415-430. Plenum Press, New York-London, 1976.

A reliable mathematical model is an indispensable tool in the studies of the influence of sources in one region on the air pollution in the surrounding regions. Assume that it is desirable to find out the influence of the sources in a sub-domain A of the space domain on the concentrations and the depositions in whole space domain. It is necessary to run the model twice: the first time with all sources in the space domain, the second time only with the sources in the sub-domain A (the emission from the sources that are outside the sub-domain A being set to zero). Then it is possible to calculate the relative parts of the concentrations and depositions caused by the sources in the sub-domain A at each grid-square (by dividing the results obtained in the second run with the corresponding results obtained in the first run). One can also calculate the percentages of the concentrations and depositions caused by the sources in sub-domain A (multiplying by 100 the relative parts of the concentrations and depositions caused by the sources in the sub-domain A calculated as described above).

Several experiments were carried out to calculate the influence of the sources from some of the big European countries to the surrounding areas have been carried out. Results concerning the contributions of German sources (in July 1989) to certain areas surrounding Germany are shown in Fig. 10.20 (the contributions in percentages of sulphur dioxide concentrations due to German sources) and Fig. 10.21 (the contributions in percentages of nitrogen dioxide concentrations due to German sources).

It should be mentioned that one should be careful when the results obtained as described above (examples being shown in Fig. 10.20 - Fig. 10.21) are to be interpreted. The results demonstrate what will happen in a rather hypothetical situations (when only the sources in one sub-domain are activated and all other sources closed). It is necessary to point out here that the results shown in Fig. 10.20 - Fig. 10.21 are telling us what would happen if only the German sources are active and nothing more. Sometimes results of this type are used to draw a conclusion that can be used to answer the question: what would happen when the German sources are closed? Consider, for example, the concentration or the deposition of any air pollutant in a grid-square where the contribution due to German sources is $p\%$. Then one concludes that the concentration or deposition in the grid-square under consideration will be decreased by $p\%$ if the German sources are closed. In general, such a conclusion is not true, because some of the processes that are involved in the model are non-linear (especially the chemical reactions). Therefore if one wants to study what would happen in the situation where the German sources are closed (instead of what is happening when only the German sources

are active, and all other sources closed), then one **must** in general run the model with the German sources closed (and all the other sources active).

However, some pollutants (as, for example, the sulphur dioxide and sulphate) participate mainly in linear processes (or, at least, the linear processes are clearly dominating the non-linear processes). Therefore, one can safely conclude (when this is the case) that if the contribution in a given grid-square (caused, say, only by German emission sources) is $p\%$, then also the result of closing the German emission sources will lead to a reduction of the appropriate concentration and depositions by about $p\%$.

Of course, the situation where all emission sources that are outside a given sub-domain A are closed (or all emission sources that are inside a given sub-domain A are closed) is an extreme case. In fact, such an example have been used in this section in order to illustrate the fact that there exists a serious danger that the results obtained in such an extreme case might be mis-interpreted (and even mis-used).

In most of the cases that arise in practice, the situation is much simpler: one wants to know what would happen if the emissions in a given sub-domain (or in the whole space domain) are reduced or increased by a certain amount. The answer to this question is quite straight-forward if one is prepared to run once more the model with the required changes of the emissions.

Sometimes one wants to use only the basic run (with all emission sources in the space domain active) in order get the answer to the question stated above. This is desirable, because very often many different reductions and/or increases of the emissions (or some of the emissions) have to be tried. It should be clear, however, that all attempts to use only the basic run are unavoidably based on some simplifying assumptions (very often, assumptions introducing linear relationships; see Chapter 1). Such simplifying assumptions may work quite satisfactorily for some air pollutants, but this is not true in general. Therefore, one should be very careful when such simplifying assumptions are used. While the computational work can be reduced very considerably by such assumptions (especially when many cases are to be investigated; and, as mentioned above, this is often the case in practice), there is a danger for obtaining wrong results (and drawing wrong conclusions), because the simplifying assumptions made are simply not valid for all air pollutants studied.

This means that if rules based on simplifying assumptions are to be used, then the modeller should make it sure that the rules have somehow been verified (for example, by running a reliable mathematical model several times for key changes of the emissions and by comparing the results obtained by the model with the results obtained by using simplifying assumptions). In the case where the result of such a verification has been satisfactorily good (and only in such a case), the simplifying assumptions could be used again and again in many situations that arise in practice.

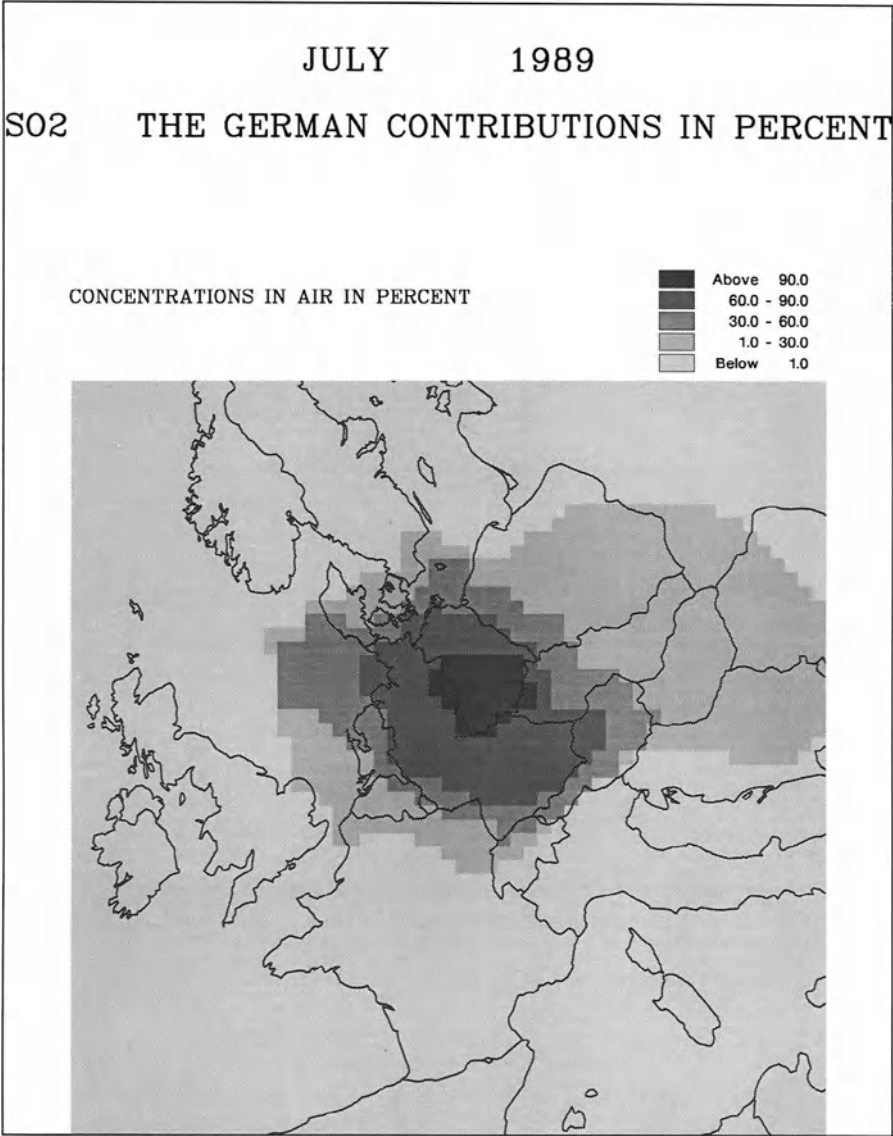


Figure 10.20
Contributions (in percent) to the sulphur dioxide concentrations
in Europe from the German emission sources.

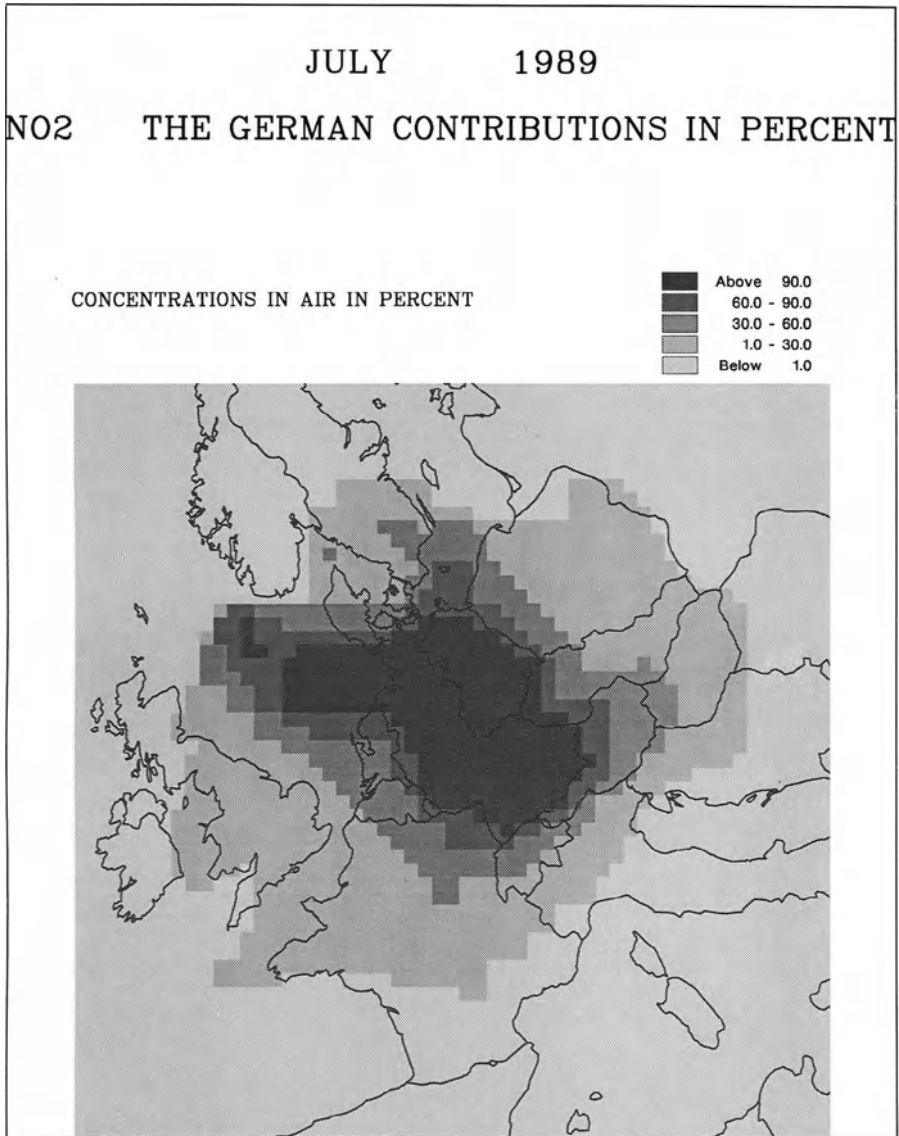


Figure 10.21
Contributions (in percent) to the nitrogen dioxide concentrations
in Europe from the German emission sources.

Simplifying assumptions have not been used in connection with **the Danish Eulerian Model**. This means that if some reduction or increase of the emission sources in a given part of the space domain (or on the whole space domain) are interesting, then the model is just run with these changes and the results (concentrations and/or depositions) obtained in such a run are compared with the corresponding results from the basic run.

Some results that are obtained by changing the emissions in the space domain (i.e. by different emission scenarios) will be presented in the next sections.

10.5. Running scenarios in which the emissions are varied

"One of the major benefits of modelling is that trends due to emission changes can be separated from meteorological changes".

Trond Iversen: *"Modelled and Measured Transboundary Acidifying Pollution in Europe - Verification and Trends"*. Atmospheric Environment, 27A(1993), p. 916.

In Chapter 1 and Chapter 2 it has been pointed out that the most important task (or, at least, one of the most important tasks) which can successfully be solved when a reliable mathematical LRTAP is available is the investigation of the influence of either decreasing or increasing the emission sources on the concentrations and depositions in the area under consideration. Several such experiments have been run by using **the Danish Eulerian Model**. Results obtained in two of these experiments will be given and briefly discussed in this section.

All European emissions of nitrogen oxides have been reduced by 25% in the first experiment. Results concerning the changes in the concentrations of sulphur dioxide, nitrogen dioxide, ozone and hydroxyl radical that are due to the above reduction of the European emissions of nitrogen oxides are presented in Fig. 10.22 - Fig. 10.26. The changes of the concentrations are obtained by comparing the concentrations that are calculated in two different runs of the model:

- (i) a basic run (where all European emissions, including here the emissions of nitrogen oxides, are kept unchanged)

and

- (ii) a second run (where all European emissions of nitrogen oxides are reduced by 25%, while the remaining emissions are kept unchanged).

Let us consider first the changes of the sulphur dioxide concentrations. Since the sulphur emissions have been kept unchanged (in both runs of the model), one should expect small changes of the sulphur dioxide concentrations in the second run. The changes, in percent, are shown in Fig. 10.22. It is seen that the changes are indeed very small. However, it is much

more important in this situation to try to evaluate the sensitivity of the model to small changes in the input data. In other words, it is important to try to give an answer to the following question:

Are the small changes shown in Fig. 10.22 (several percent only) caused by errors due to the numerical algorithms used in the model (truncation errors) and/or by errors due to the use of computer arithmetic (rounding errors)?

It is difficult to give a direct answer to this question by studying only the results in Fig. 10.22. However, a careful analysis of the processes involving sulphur dioxide **indicates** that the answer of the question is negative (i.e. neither the truncation errors nor the rounding errors are causing the small changes; the changes are caused by the change of the emissions of nitrogen oxides). If this indication can be strictly confirmed, then an important conclusion can be drawn: both the truncation errors and the rounding errors are much smaller than the errors caused by uncertainties in the input data and the errors due to uncertainties in the implementation of the physical processes in the model. Therefore some further analysis of this situation seems to be necessary.

The sulphur dioxide participates in only **two** of the chemical reactions implemented in the model. The first reaction, in which sulphate is produced from sulphur dioxide (see reaction 64 in Table 2.3 in Chapter 2), is not very important in connection with the above conclusion. The second reaction, in which sulphur dioxide reacts with the hydroxyl radical (producing again sulphate; see reaction 63 in Table 2.3), is more important for the analysis of this particular situation. The key point is that the concentrations of the hydroxyl radical depend rather strongly on the emissions of nitrogen oxides. The changes of the concentrations of the hydroxyl radical due to the reduction of the emissions of nitrogen oxides are shown in Fig. 10.26. The main part of the sulphate concentrations is produced by the first reaction. Nevertheless, it is clearly seen (by comparing the results in Fig. 10.22 with those in Fig. 10.26) that the changes of the sulphur dioxide concentrations (in percent) are smaller in the parts of the space domain where the changes of the concentrations of the hydroxyl radical (again in percent) are larger and vice versa.

The effect is opposite when the changes of the sulphate concentrations are studied; see Fig. 10.23. The changes of the sulphate concentrations (in percent) are larger when the changes of the concentrations of the hydroxyl radical (again in percent) are also larger. The changes of the sulphate concentrations are again small, but these are in general larger than the changes of the sulphur dioxide concentrations (in some parts of the space domain by a factor larger than two).

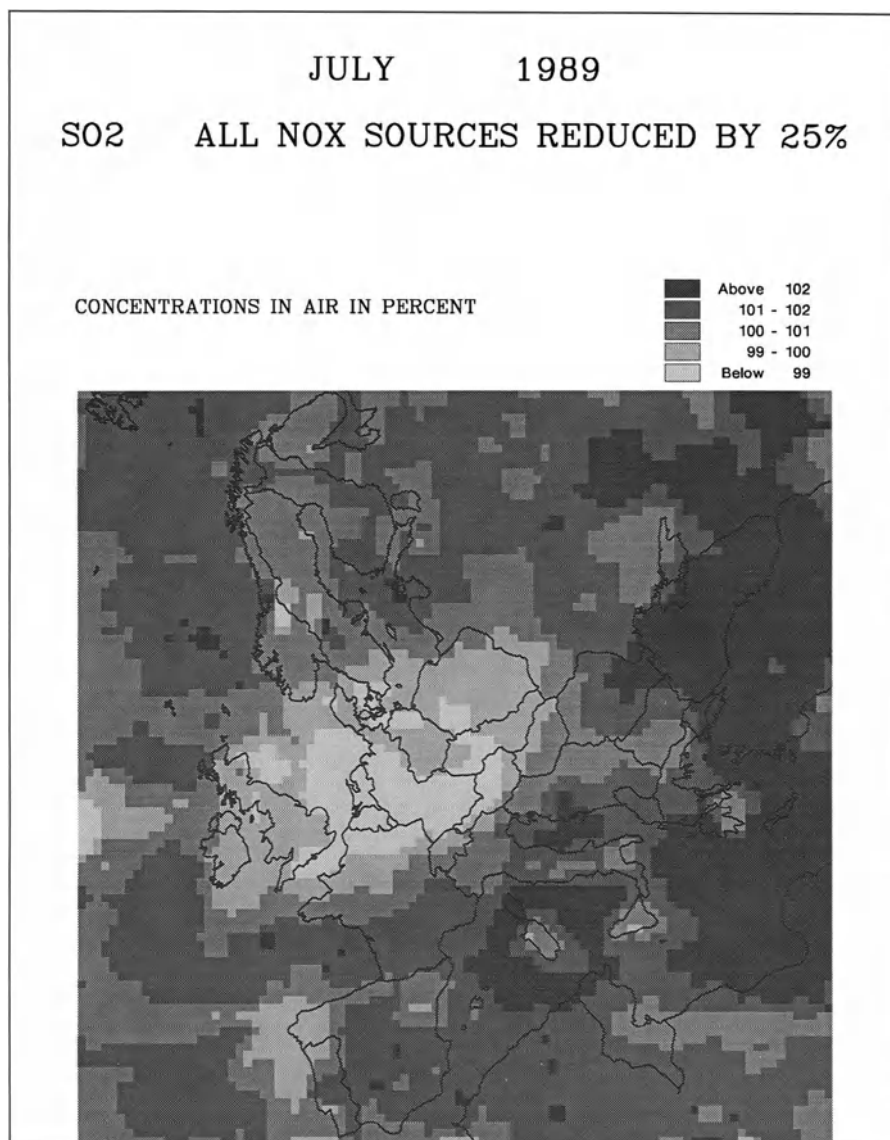


Figure 10.22
Changes (in percent) of the sulphur dioxide concentrations
when the NO_x emissions in Europe are reduced by 25%.

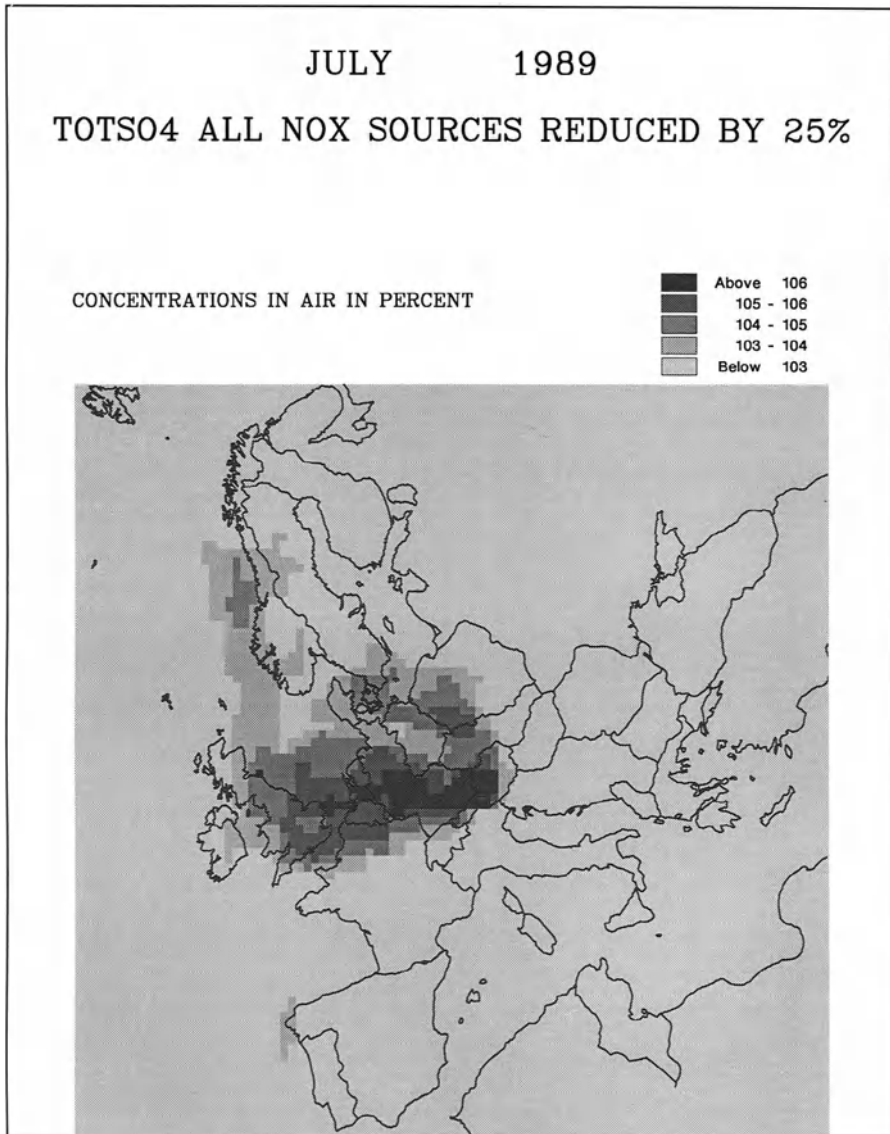


Figure 10.23
Changes (in percent) of the sulphate concentrations
when the NO_x emissions in Europe are reduced by 25%.

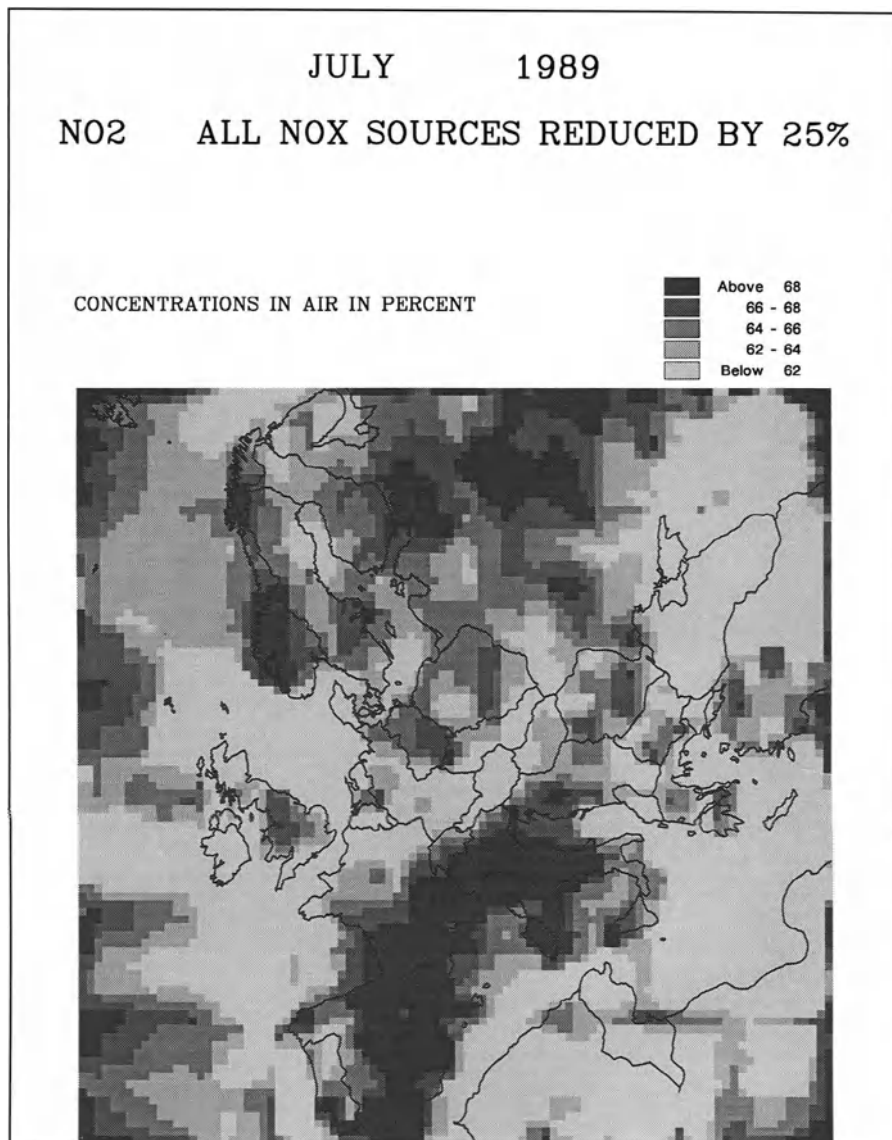


Figure 10.24
Changes (in percent) of the nitrogen dioxide concentrations
when the NO_x emissions in Europe are reduced by 25%.

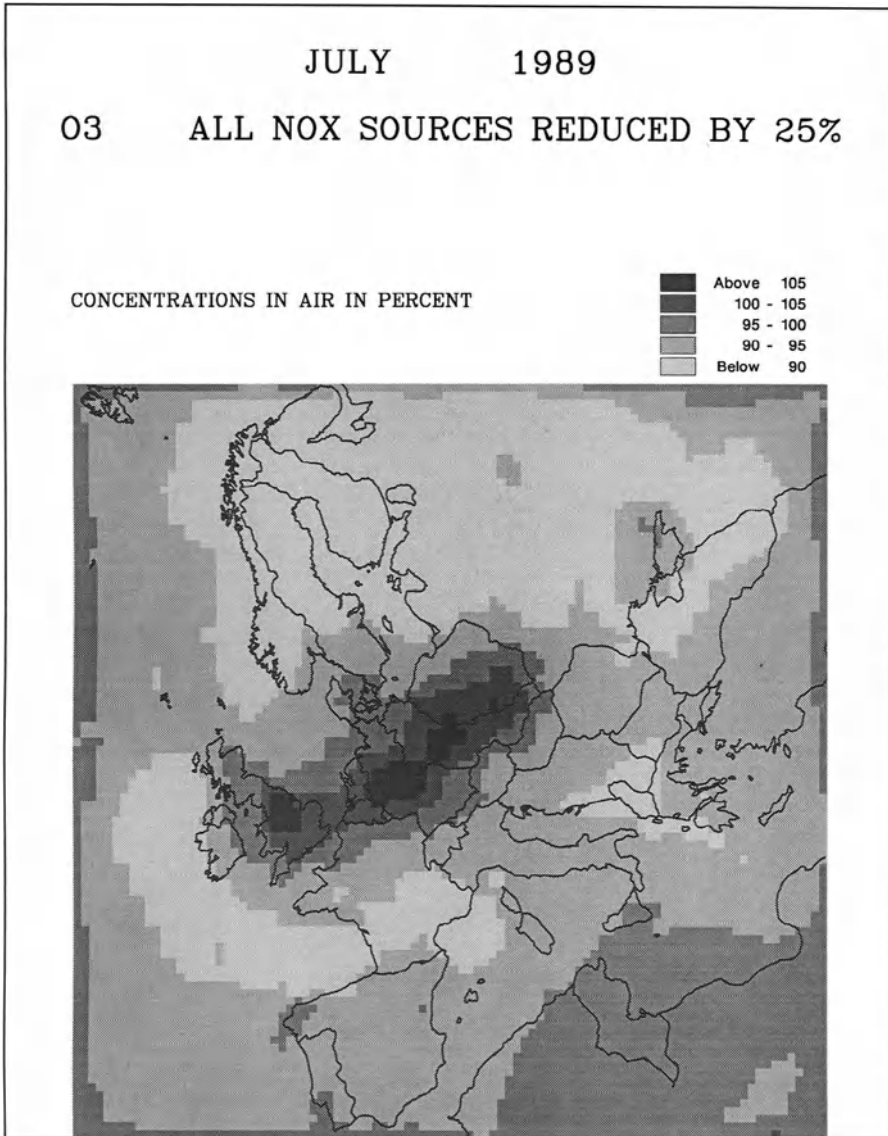


Figure 10.25
Changes (in percent) of the ozone concentrations
when the NO_x emissions in Europe are reduced by 25%.

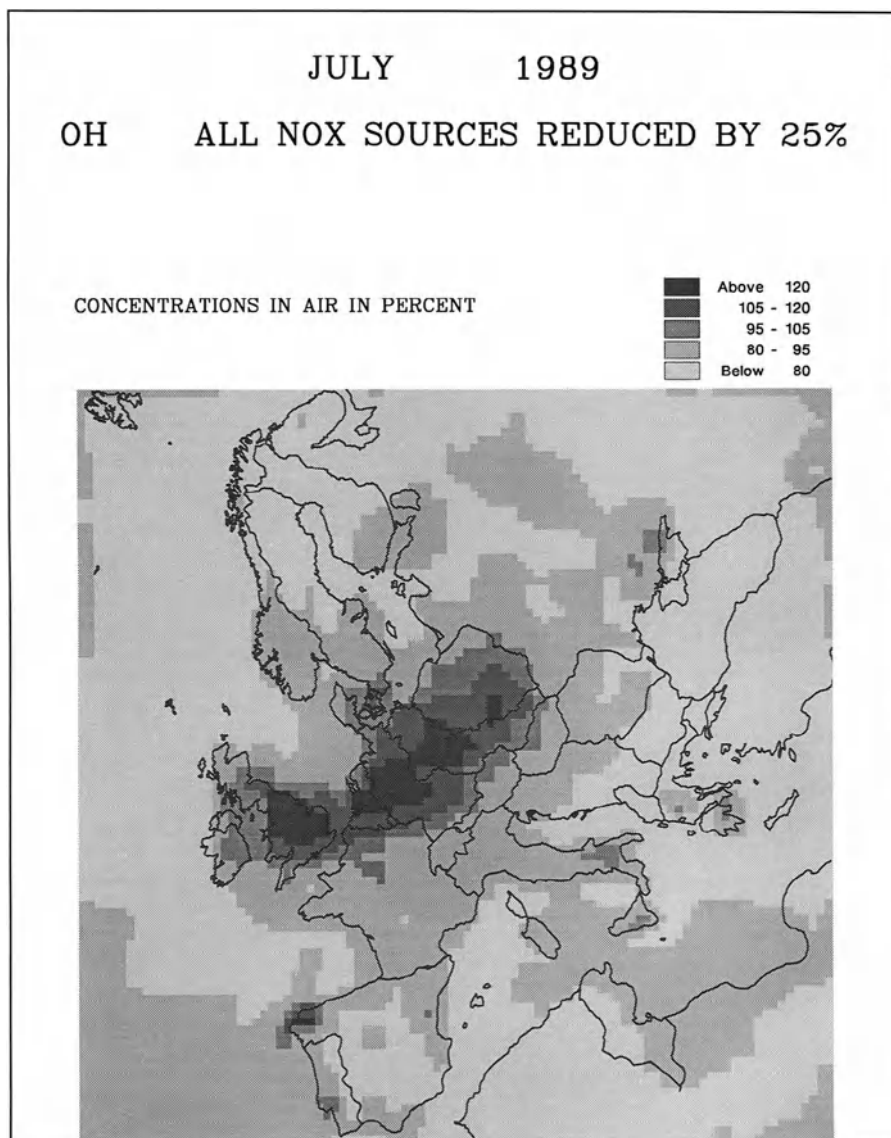


Figure 10.26
Changes (in percent) of the hydroxyl radical concentrations
when the NO_x emissions in Europe are reduced by 25%.

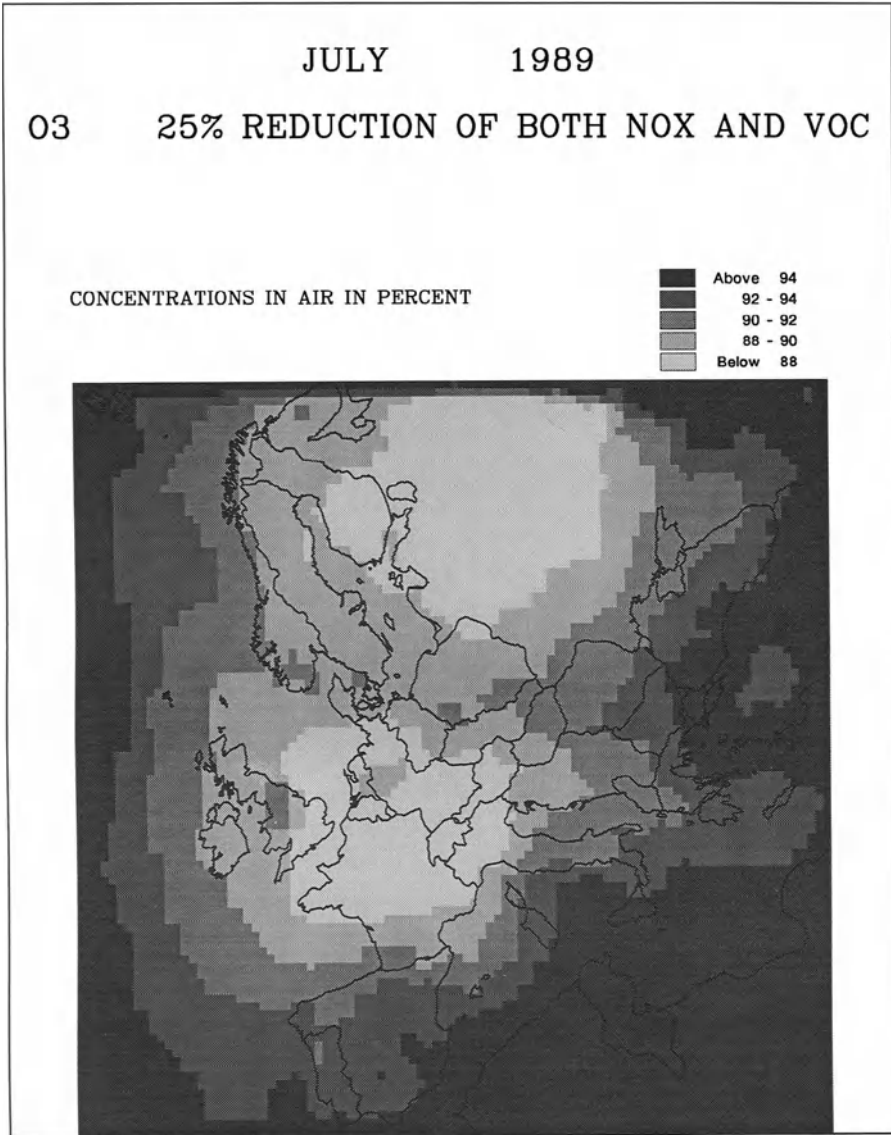


Figure 10.27
Changes (in percent) of the ozone concentrations
when the NO_x and VOC emissions in Europe are reduced by 25%.

Let us reiterate that the above analysis (together with the tests performed in Chapter 4) could be considered as an indirect indication that the numerical algorithms used in the model are sufficiently accurate (in comparison with the uncertainties of the input data and with the uncertainties of the physical mechanisms implemented in the model). However, it should also be stressed that this is only an indication and much more experiments and analyses of different situations are necessary in order to obtain a more decisive arguments that such a conclusion is really correct.

The changes of the nitrogen dioxide concentrations (when all European emissions of nitrogen oxides are reduced by 25%) are given in Fig. 10.24. The greatest changes are mainly achieved in the Central and Western Europe (where there are the biggest sources of emissions of nitrogen oxides) and over the Atlantic Ocean, the Mediterranean, Africa and Asia (where there are no emission sources or, at least, there are only some small emission sources of nitrogen oxides). In all other parts of the space domain the changes are more moderate.

It is also seen that the processes of forming nitrogen dioxide are not linear. If these processes were linear, then all nitrogen dioxide concentrations should be reduced to about 75% of the basic levels of the nitrogen dioxide concentrations when the emissions of nitrogen oxides are reduced by 25%. The results show that all concentrations are reduced by more than 30% (and in some parts of the space domain by 35% - 40%).

The changes of the ozone concentrations (in percent), which correspond to the reduction of all European emissions of nitrogen oxides by 25%, are shown in Fig. 10.25. It is seen that in some parts of the space domain the reduction of the emissions of the nitrogen oxides leads to an increase of the ozone concentrations. It is also seen that this happens in the highly industrialized regions of Europe: Central England and parts of Belgium, Netherlands, Germany, Poland and Checkia. This results shows very clearly that it is difficult to predict the behaviour of the changes of the ozone concentrations as a function of changes of the emissions of nitrogen oxides. The relationship that describes the variation of the ozone concentrations when the emissions of the nitrogen oxides are varied is not simple. In fact, the opposite is true; this relationship is very complicated and many other factors are to be taken into account if a reliable prediction of the influence of variations of the emissions of nitrogen oxides on the variations of the ozone concentrations is to be calculated without running the model with the changed emissions.

The results that are shown in Fig. 10.25 indicate that it is not sufficient to reduce the emissions of the nitrogen oxides if the ozone concentrations are to be reduced. One should reduce both the emissions of nitrogen oxides and the emissions of the anthropogenic VOC's in order to achieve some reduction of the ozone concentrations. This is demonstrated by the second experiment where both the emissions of nitrogen oxides and the emissions of the anthropogenic VOC's are reduced by 25%. The changes of the ozone concentrations in this situation are shown in Fig. 10.27. It is seen that the ozone concentrations are reduced in the whole space domain. The reductions in Central and Western Europe and Southern England are greatest. However, the reductions of the ozone concentrations are not very impressive; even in Central and Western Europe and in Southern England. This means that the two experiments demonstrated here seem to indicate that a significant reduction of the monthly mean values of the ozone concentrations in Europe can be achieved only by reducing very

considerably both the emissions of nitrogen oxides and the emissions of the anthropogenic VOC's.

Some experiments performed in Zlatev et al. [538,539] indicate that while the monthly mean values of the ozone concentrations in Europe seem to be rather insensitive to changes of emissions of nitrogen oxides and/or emissions of anthropogenic VOC's, the periods with high ozone concentrations are reduced more considerably. This is an important fact, because the periods with high ozone concentrations are (or, at least, can be) dangerous for plants, animals and humans.

The sensitivity of the ozone concentrations to changes of emissions of nitrogen oxides and/or emissions of anthropogenic VOC's is an important issue and much more research in this direction seems to be necessary. This is especially true for the periods with high ozone concentrations.

The hydroxyl radical is a very active chemical agent. It reacts with many other pollutants. Therefore the levels of the concentrations of the hydroxyl radical should also be studied carefully; the levels of these concentrations have influence on the levels of the concentrations of many of the other pollutants. It is important to investigate the sensitivity of the concentrations of the hydroxyl radical to changes of different emissions. The knowledge obtained during such investigations may be used to explain the behaviour of the variations of many other pollutants that are caused by different changes of the emissions. One example has already been given above (see the discussion concerning the changes of the sulphur dioxide concentrations that are caused by the reduction of the emissions of nitrogen dioxide).

The changes of the concentrations of the hydroxyl radical that are caused by the reduction of all European emissions of nitrogen dioxides by 25% are shown in Fig. 10.26. It is seen that the changes vary in a quite large interval (most of the changes are in the range between 80% and 120%, but some changes may be less than 80% or greater than 120%). The concentrations of the hydroxyl radical are increased (when the emissions of nitrogen oxides are decreased) in Central and Western Europe and England; while in most of the other parts of the space domain the concentrations of the hydroxyl radical are decreased.

The changes of the concentrations of the hydroxyl radical that are caused by reducing both the emissions of nitrogen oxides and the emissions of VOC's are quite similar to those shown in Fig. 10.26. However, the changes do not vary in so large interval in the latter case (most of the changes are in the range from 90 % to 105%, but some of the changes may be less than 90% or greater than 105%).

The changes of the concentrations of many other pollutants were also studied. As an illustration of this statement, let us mention that the behaviour of the variation of the concentrations the HO_2 radical is quite similar to the behaviour of the variation of the concentrations of the hydroxyl radical (and, moreover, this is true both in the case where only the emissions of nitrogen oxides are reduced and in the case where both the emissions of the nitrogen oxides and the emissions of VOC's are reduced).

It must be emphasized here that the results in this section are given in order to show how the models can be used. As mentioned several times above, the results indicate that the

numerical methods used in the particular model used in these experiments, **the Danish Eulerian Model**, are sufficiently accurate (which implies that even very small changes of the concentrations and/or the depositions can clearly be detected by the model). However, one should be careful not to draw too many and too strong conclusions. It must also be emphasized that much more experiments are needed (and, perhaps some further development of the model is necessary) in order to confirm some trends and/or to establish other trends in the variations of different concentrations and deposition when some emission sources are varied.

10.6. Studying episodes with high ozone concentrations

"Both changes in emissions of volatile organic compounds (VOC) and NO_x are important for changes of episodic ozone."

Øystein Hov: *"Model Calculations of Ozone in the Atmospheric Boundary Layer in Europe"*. In: **"ATMOSPHERIC OZONE RESEARCH AND ITS POLICY IMPLICATIONS"** (T. Schneider, S. D. Lee, G. J. R. Wolters and L. D. Grant, eds.), pp. 657-666, Elsevier Science Publishers B.V., Amsterdam, 1989.

High ozone concentrations can be dangerous for plants, animals and humans. The ozone concentrations depend on the emissions of both nitrogen oxides and volatile organic compounds (the emissions of NO_x and VOC). However, it has been shown in the previous section that the reductions of the emissions of NO_x and VOC must be very considerable in order to obtain a significant reduction of the mean ozone concentrations (on monthly and daily bases) in the troposphere. Therefore it is important to study carefully the high concentrations of ozone (and especially the time variation of the high ozone concentrations as well as the duration of periods where the ozone concentrations are high).

The ozone concentrations have a clear diurnal variation: these are normally highest during the afternoon and lowest during the night. The diurnal variation of the ozone concentrations at some of the EMEP measurement stations is illustrated in Zlatev et al. [538].

A careful study of the variation of the daily maxima of the ozone concentrations at several EMEP measurement stations in different European countries shows that there are short periods (which are called, as mentioned above, episodes) when the maximal daily values of the ozone concentrations are rather high.

Results that are presented in Zlatev et al. [538] show clearly that there were three such periods, on the parts of Central and Western Europe, in July 1985 (1-6 July, 11-15 July and 23-27 July). The same episodes have also been detected, by using measurements taken at different EMEP stations, in Grennfelt et al. [216].

Some episodes in June 1989 have been studied in Zlatev et al. [539]. These episodes were again in the Central and Western Europe.

A few results, which represent temporal variations of calculated and measured ozone concentrations (in July 1989), are shown in Fig. 10.28 and in Fig. 10.29.

The time variations of the calculated and measured ozone concentrations at Deuselbach in Germany are shown in Fig. 10.28. This plot indicates that there are two relatively long episodes in the part of the space domain around Deuselbach; in the periods 5-13 July 1989 and 22-30 July 1989.

The time variations of the calculated and measured ozone concentrations at Ispra in Italy are shown in Fig. 10.29. The model results do not indicate a clear episode in the region around Ispra in July 1989. However, the ozone concentrations are about 80 ppb the whole time, and such a level of the ozone concentrations is considered by many scientists as rather high.

10.7. Studying the concentrations and depositions due to Danish sources

"The main objective of EMEP is to provide governments with information on deposition and concentration of air pollutants, as well as on the quantity and significance of pollutant fluxes across national boundaries".

Anton Eliassen and Jørgen Salbones: *"Modelling of Long-range Transport of Sulphur over Europe: A Two-year Model Run and Some Model Experiments"*. Atmospheric Environment, **17**(1983), pp. 1457-1473.

As mentioned several times, the reduction of the air pollution is normally an expensive procedure. It would be very difficult for the small European countries to reduce the concentrations and the depositions of the dangerous air pollutants to an acceptable level. The main reason for this is the fact that a great part of the pollution in such small countries is due to transport from other countries. This explains why cooperative actions are necessary (this topic has also been discussed in Chapter 1).

It is important to investigate carefully the influence of the sources in a given small country on the concentrations and the depositions both in the country itself and in its surroundings. Some results that have been obtained in the case where the country under consideration is Denmark will be presented in this section.

The contributions for 1989, in percent, of the Danish emission sources to the concentrations of the sulphur dioxide in Denmark and its surrounding are shown in Fig. 10.30. It is seen that in many parts of Denmark the contributions of the Danish sources to the sulphur dioxide concentrations are under 50% (this means that, roughly speaking, more than one half of the concentrations are caused by emission sources that are outside Denmark). However, it is also seen that the contribution of the Danish sources to the sulphur dioxide concentrations in the Southern parts of Sweden is rather considerable.

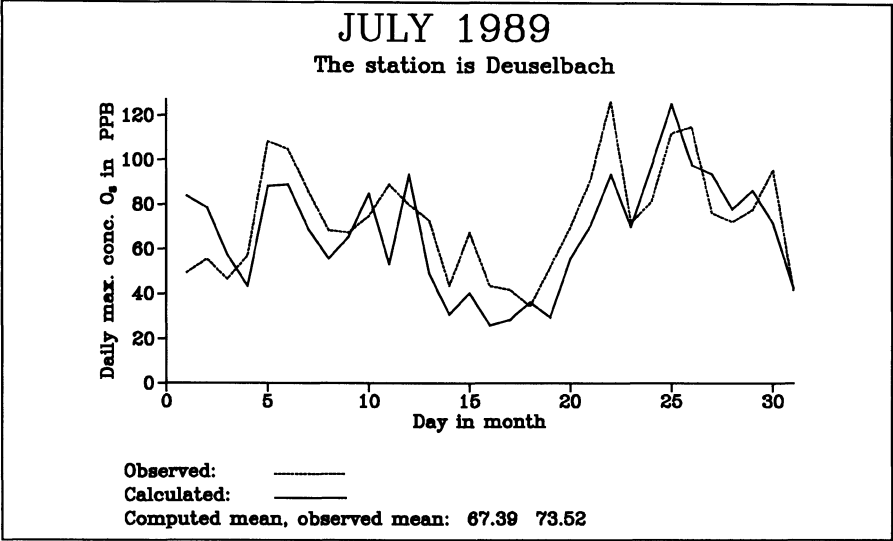


Figure 10.28
Comparison of daily maxima of calculated and measured ozone concentrations
in July 1989 at Deuselbach (Germany).

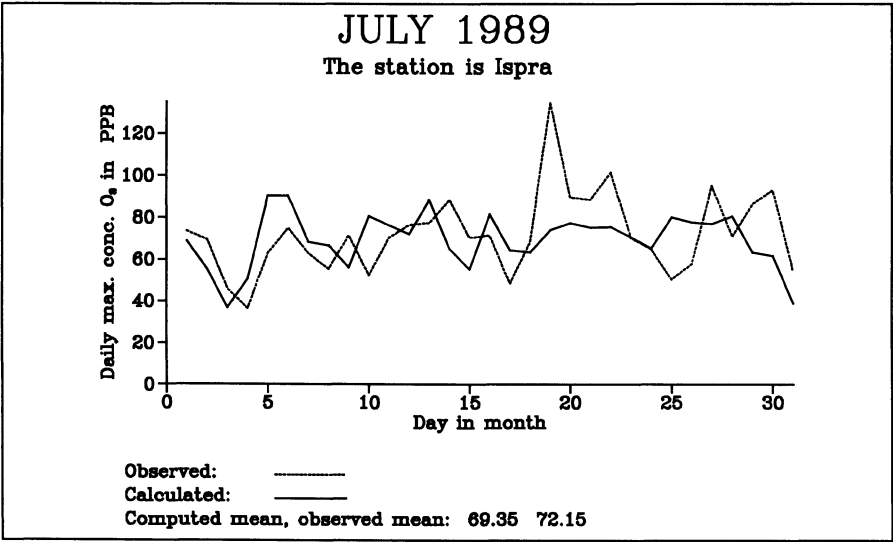


Figure 10.29
Comparison of daily maxima of calculated and measured ozone concentrations
in July 1989 at Ispra (Italy).

The corresponding contributions in the case where nitrogen dioxide concentrations are studied instead of sulphur dioxide concentrations are shown (again in percent) in Fig. 10.31. The main pattern is quite similar:

(i) a big part of the nitrogen dioxide concentrations in Denmark are caused by sources that are outside Denmark

and

(ii) the contribution of the Danish sources to the nitrogen dioxide concentrations in Southern parts of Sweden is rather considerable.

It is also seen from Fig. 10.31 that the nitrogen dioxide concentrations are transported on longer distances than the sulphur dioxide concentrations. There are several different reasons by which such a phenomenon can be explained. The chemical reactions outside Denmark are not very effective, because all emission sources there are closed. This is probably the most important reason. The fact that the dry deposition coefficient for nitrogen dioxide is considerably less than the dry deposition coefficient for sulphur dioxide is another reason for this difference. It should also be mentioned that the nitrogen dioxide is not deposited by precipitation, while the wet deposition coefficient for sulphur dioxide is considerably large. More details about the deposition coefficients used for the different air pollutants that are involved in **the Danish Eulerian Model** can be found in Harrison et al. [228] or Zlatev et al. [538,539].

10.8. Concluding remarks and references concerning simulations with large air pollution models

"Thus, every attempt is made to incorporate all relevant processes in as much detail as possible. However, our ability to do so is limited by three factors. Firstly, we do not as yet have a complete understanding of all the relevant processes. Secondly, the inclusion of the detailed descriptions of all the processes (even if we understood them) is likely to strain the resources of the best available computers. The third reason for being less than comprehensive is related to the need to test the model against observations".

A. Venkatram, P. K. Karamchandani and P. K. Misra: *"Testing a Comprehensive Acid Deposition Model"*. Atmospheric Environment, **22**(1988), 737-747.

Several experiments with **the Danish Eulerian Model** have been described in this chapter. This model has been developed at the National Environmental Research Institute (Roskilde, Denmark). The efforts to improve both the robustness and the reliability of the model are continuing by including new and more advanced modules in it. **The Danish Eulerian Model** is described in Harrison et al. [228] and in Zlatev et al. [538,539].

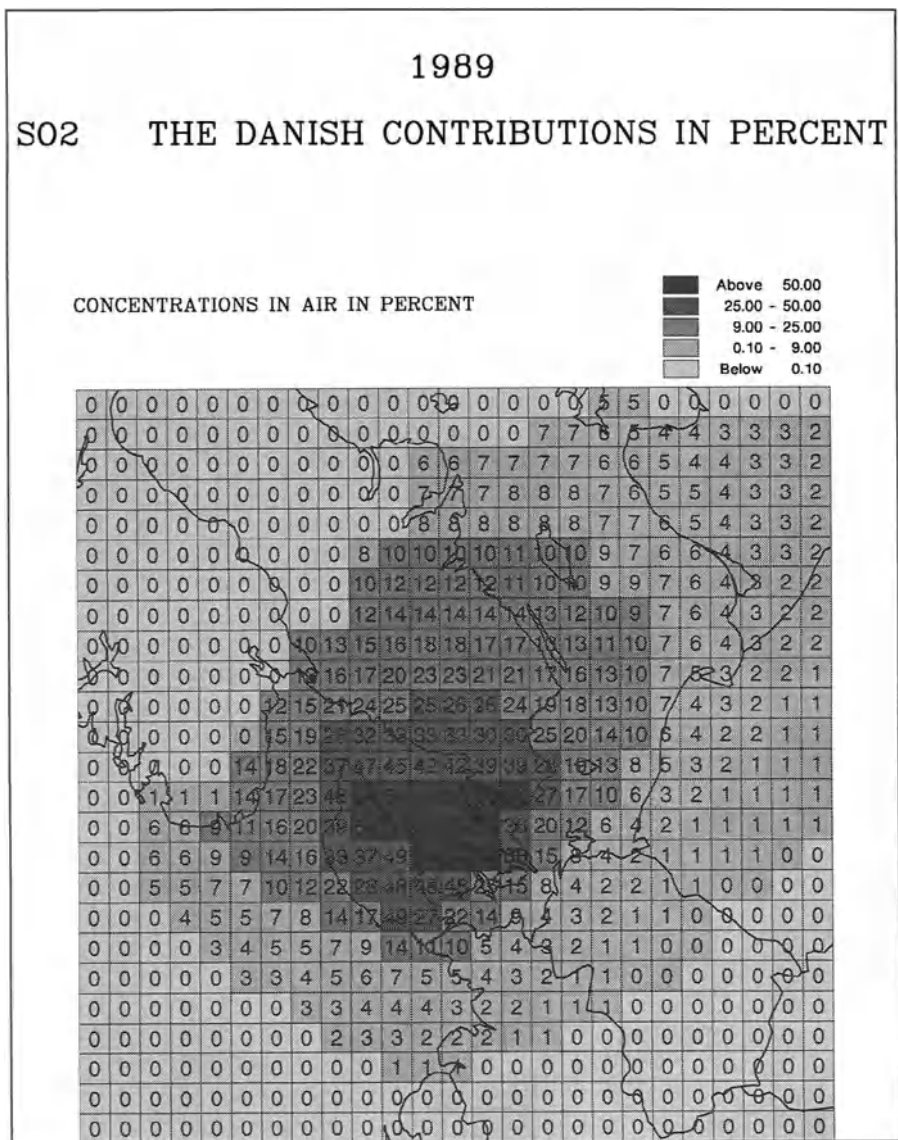


Figure 10.30
The contributions (in percent) of the Danish sources
to the sulphur dioxide concentrations
in the region surrounding Denmark for 1989.

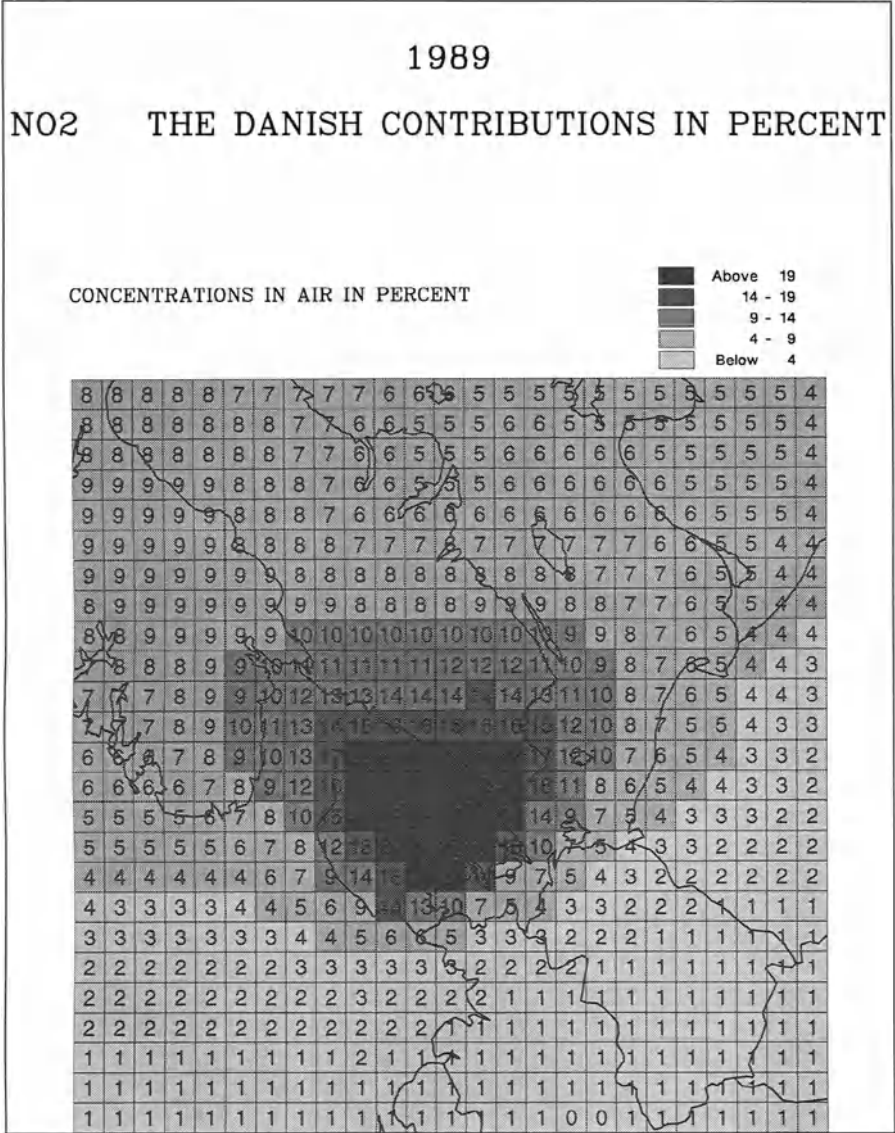


Figure 10.31
The contributions (in percent) of the Danish sources
to the nitrogen dioxide concentrations
in the region surrounding Denmark for 1989.

An attempt to locate the position of **the Danish Eulerian Model** among the other models in the set of the LRTAP models has been made in **Section 10.1**. It is emphasized there, in **Section 10.1**, that the main aim is not to upgrade the model to a comprehensive model (some examples of comprehensive models are described and discussed in Venkatram et al. [459], Chang et al. [66], Ebel et al. [144] and Hass et al. [230,231]; more examples can be given), but rather to preserve the possibility of running the model over long time-intervals and, thus, to preserve the possibility of studying also seasonal variations of the concentrations and the depositions of various air pollutants that are involved in the model. However, by adding new and more complicated modules to the model and by exploiting extensively high-speed computers, the gap between **the Danish Eulerian Model** and the comprehensive LRTAP models becomes smaller and smaller.

Concentration levels in Europe have been discussed in **Section 10.2**. These concentration levels have been obtained by using the new version of **the Danish Eulerian Model**. The spatial resolution of the new version is much better than that of the old version (in fact, the spatial resolution has been improved by a factor of nine; 9216 grid-points are used in the new version instead of the 1024 grid-points used in the old version that is described in Harrison et al. [228] and in Zlatev et al. [538,539]).

The need of comparisons of the model results (both concentrations and depositions) with measurements as well as some comparisons with measurements are discussed in **Section 10.3**. Both scatter plots and temporal variations of measured and calculated concentrations are presented in **Section 10.3**.

The influence of emission sources in one part of the space regions on the air pollution in the surrounding parts is studied in **Section 10.4**. Some results concerning the influence of German emission sources on the air pollution of Germany and its surroundings are presented. It should be mentioned that the experiment that has been carried out in this section was performed only in order to demonstrate that the model is able to solve such tasks, and not to produce some results that can be used for any practical purposes. The results are for July 1989. One should run the model on much longer time-intervals (as, for example, for the whole 1989) in order to obtain more representative results concerning the concentrations and the depositions that are to be studied.

Some runs with different emissions (different "scenarios") have been presented in Zlatev et al. [538,539]. Results that have been obtained with different scenarios are given in **Section 10.5**. An analysis of the results is also presented in **Section 10.5**.

The importance of studying ozone concentrations, and especially high ozone concentrations, is discussed in **Section 10.6**. Some results, which have been obtained in the comparisons of observed and calculated daily maxima of the ozone concentrations at two EMEP stations, are also given in **Section 10.6**.

The contributions of the Danish emission sources to the concentrations and depositions of air pollutants in Denmark and in the surrounding of Denmark are discussed in **Section 10.7**. There are two differences between the results presented in **Section 10.4** and the results presented in **Section 10.7**.

The first difference is the time-interval used in the runs. The results that are presented in Section 10.7 have been obtained by running the model for the whole 1989, while the corresponding results in Section 10.4 are only for one summer month (July 1989). This means that it should be expected the results in Section 10.7 to be much more representative.

The second difference is the size of the country. Denmark is a much smaller country than Germany. This is why the contributions of sources that are outside Denmark to the concentrations and depositions in Denmark are, in general, much greater than the contributions of sources that are outside Germany to the concentrations and depositions in the central parts of Germany. This conclusion explains why

the small countries are (and must be) much more interested in performing common international actions in the efforts to reduce the air pollution to acceptable levels in an optimal way.

This topic, the topic of reducing the air pollution to acceptable levels in a cheap and reliable way, has also been discussed in Chapter 1.

REFERENCES

1. **S. Abarbanel and D. Gottlieb:** *"Higher order accuracy finite difference algorithms for quasi-linear, conseervation law hyperbolic systems"*. Math. Comp., **27** (1973), 505-424.
2. **S. Abarbanel, D. Gottlieb and E. Turkel:** *"Difference schemes with fourth order accuracy for hyperbolic equations"*. SIAM J. Appl. Math., **29** (1975), 329-351.
3. **C. A. Addison, W. H. Enright, P. W. Gaffney, I. Gladwell and P. M. Hanson:** *"A decision tree for the numerical solution of initial value ordinary differential equations"*. Report CCS 86Z3. Centre for Computer Science, Chr. Michelsen Institute, Fantoftvegen 38, N-5036 Fantoft, Bergen, Norway, 1986.
4. **J. Alcamo, L. Hordijk, J. Kamari, P. Kauppi, M. Posch and E. Runca:** *"Integrated analysis of acidification in Europe"*. Report. International Institute for Applied Systems Analysis (IIASA), A-2361 Laxenburg, Austria, 1984.
5. **J. Alcamo, P. Kauppi, M. Posch and E. Runca:** *"Acid rain in Europe: A framework to assist decision makers"*. Report WP-84-32. International Institute for Applied Systems Analysis (IIASA), A-2361 Laxenburg, Austria, 1984.
6. **R. Alexander:** *"Diagonally implicit Runge-Kutta methods or stiff ODE's"*. SIAM J. Numer. Anal., **14** (1977), 1006-1021.
7. **R. Alexander and J. J. Coyle:** *"Runge-Kutta methods for differential- algebraic systems"*. SIAM J. Numer. Anal., **27** (1990), 736-752.
8. **E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen:** *"LAPACK: Users' guide"*. SIAM, Philadelphia, 1992.
9. **A. Arakawa:** *"Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow, Part I"*. J. Comput. Phys., **1** (1966), 119-143.
10. **W. E. Arnoldi:** *"The principle of minimized iteration in the solution of the matrix eigenvalue problem"*. Quart. Appl. Math., **9** (1951), 17-29.
11. **W. A. Asman:** *"Ammonia emissions in Europe: Updated emission and emission variations"*. Report 228471008, RIVM, Bilthoven, The Netherlands, 1990.
12. **W. A. Asman and A. J. Jansen:** *"A long-range transport model for ammonia and ammonium for Europe and some model experiments"*. Report No. R-1986-6. Institute for Meteorology and Oceanography, State University of Utrecht. Petten, The Netherlands, 1986.

13. **W. A. Asman and A. J. Jansen:** *"A long-range transport model for ammonia and ammonium for Europe"*. Atmos. Environ., **21** (1987), 2099-2119.
14. **O. Axelsson:** *"A generalized SSOR method"*. BIT, **12** (1972), 443-467.
15. **O. Axelsson:** *"On preconditioned conjugate gradients method"*. In: **"CONJUGATE GRADIENT METHODS AND SIMILAR TECHNIQUES"** (I.S.Duff, ed.), pp. 23-55. Report No. R9636, A.E.R.E., Harwell, England, 1979.
16. **O. Axelsson:** *"Conjugate type methods for unsymmetric and inconsistent systems of linear equations"*. Lin. Alg. Appl., **29** (1980), 1-16.
17. **O. Axelsson:** *"A survey of preconditioned iterative methods for linear systems of algebraic equations"*. BIT, **25** (1985), 166-187.
18. **O. Axelsson and V. A. Barker:** *"Finite element solution of boundary value problems"*. Academic Press, Orlando, Florida, USA, 1984.
19. **O. Axelsson and I. Gustavsson:** *"A modified upwind scheme for convective transport equations and the use of a conjugate gradient method for the solution of non-symmetric systems of equations"*. J. Inst. Math. Applics., **23** (1979), 321-337.
20. **K. A. Bagrinowski and C. K. Godunov:** *"Difference schemes for multidimensional problems"*. Dokl. Akad. Nauk USSR, **115** (1957), 431-433.
21. **V. B. Balakin:** *"Method of Runge-Kutta type for gas-dynamics"*. U.S.S.R. Comput. Math. and Math. Phys., **10** (1970), 208-216.
22. **J. A. Balko and L. K. Peters:** *"A modeling study of SO_x - NO_x - hydrocarbon plumes and their transport to the background troposphere"*. Atmos. Environ., **17** (1983), 1965-1978.
23. **V. A. Barker:** *"Numerical solution of sparse singular systems of equations arising from ergodic Markov chains"*. Stochastic Models, **5** (1989), 335-381.
24. **A. Bayliss, D. Gottlieb, B. J. Matkowsky and M. Minkoff:** *"An adaptive pseudo-spectral method for reaction diffusion problems"*. Report No. ANL/MCS-TM-100. Mathematics and Computer Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439, USA, 1987.
25. **K. H. Becker, J. Löbel, U. Schurath, G. von Nieding, H. M. Wagner, R. Guderian, D. T. Tingey, R. Rabe and H. Schreiber:** *"Luftqualitätskriterien für photochemische oxidanten"*. Erich Schmidt Verlag, Berlin, 1983.
26. **A. Beguelin, J. Dongarra, A. Geist and V. Sunderam:** *"Visualization and debugging in a heterogeneous environment"*. IEEE Computer, Vol. **26**, No. 6, 1993, 88-95.

27. **R. Berkovicz and L. P. Prahm:** *"Pseudospectral dispersion modelling"*. In: **"PROCEEDINGS OF THE 8TH INTERNATIONAL TECHNICAL MEETING ON AIR POLLUTION MODELLING AND ITS APPLICATIONS"**, pp. 635-657. Louvain-La-Neuve, Belgium, 1977.
28. **R. Bermejo:** *"On the equivalence of semi-Lagrangian interpolation schemes and particle-in cell finite element methods"*. Monthly Weather Review, **118** (1990), 979-987.
29. **H. Bildsøe, J. P. Jacobsen and K. Schaumburg:** *"Application of density matrix formalism in NMR spectroscopy. I. Development of a calculation scheme and some simple examples"*. J. Magn. Resonance, **23** (1976), 7-151.
30. **Å. Björck and Z. Zlatev:** *"Exploiting the separability in the solution of linear ordinary differential equations"*. Comp. Math. Appl., **18** (1989), 421-438.
31. **D. L. Book, J. P. Boris and K. Hain:** *"Flux-corrected transport II: Generalizations of the method"*. J. Comp. Phys., **18** (1975), 248-283.
32. **J. P. Boris and D. L. Book:** *"Flux-corrected transport. SHASTA, a fluid transport algorithm that works"*. J. Comp. Phys., **11** (1973), 38-69.
33. **J. P. Boris and D. L. Book:** *"Flux-corrected transport III. Minimal error FCT algorithms"*. J. Comp. Phys., **20** (1976), 397-431.
34. **A. Bott:** *"A positive definite advection scheme obtained by nonlinear renormalization of the advective fluxes"*. Monthly Weather Review, **117** (1989), 1006-1015.
35. **A. Bott:** *"Reply to a comment on 'A positive definite advection scheme obtained by nonlinear renormalization of the advective fluxes'"*. Monthly Weather Review, **117** (1989), 2633-2636.
36. **J. W. Bottenheim, K. A. Bruce and K. G. Anlauf:** *"Discussion of a Lagrangian trajectory model describing long-range transport of oxides of nitrogen, the incorporation of PAN in the chemical mechanism and supporting measurements of PAN and nitrate species at rural sites in Ontario"*. Atmos. Environ., **18** (1984), 2609-2619.
37. **R. K. Brayton and C. C. Conley:** *"Some results on the stability and instability of the backward differentiation methods with non-uniform steps"*. In: **"TOPICS IN NUMERICAL ANALYSIS"** (J. H. Miller, ed.), pp. 13-33. Academic Press, New York, 1972.
38. **E. O. Brigham:** *"The Fast Fourier Transform"*. Prentice-Hall, Englewood Cliffs, 1974.
39. **R. Brodzinsky, B. K. Cantrell, R. M. Endlich and C. M. Bhumralkar:** *"A long-range transport model for Eastern North America II. Nitrogen oxides"*. Atmos. Environ., **18** (1984), 2361-2366.

40. **P. K. Brown and A. C. Hindmarsh:** *"Matrix-free methods for stiff systems of ODE's"*. SIAM J. Numer. Anal., **23** (1986), 610-638.
41. **P. K. Brown and A. C. Hindmarsh:** *"Reduced storage matrix methods in stiff ODE systems"*. Report No. UCRL 95088. Lawrence Livermore National Laboratory, Livermore, California, 1986.
42. **P. K. Brown and A. C. Hindmarsh:** *"Reduced storage matrix methods in stiff ODE systems"*. Applied Mathematics and Computations, **31** (1989), 40-91.
43. **P. J. H. Builtjes:** *"The use of emission data bases in air pollution dispersion modelling"*. In: **"AIR POLLUTION MODELLING AND ITS APPLICATIONS VII"** (H. van Dop, ed.), pp. 3-18. Plenum Press, New York, 1989.
44. **R. Bulirsch and J. Stoer:** *"Numerical treatment of ordinary differential equations by extrapolation methods"*. Numer. Math., **8** (1966), 1-13.
45. **K. Burrage:** *"A special family of Runge-Kutta methods for solving stiff differential equations"*. BIT, **18** (1978), 21-41.
46. **K. Burrage:** *"The dichotomy of stiffness: pragmatism versus theory"*. Applied Mathematics and Computations, **31** (1989), 92-111.
47. **K. Burrage and J. C. Butcher:** *"Stability criteria for implicit Runge-Kutta methods"*. SIAM J. Numer. Anal., **16** (1979), 46-57.
48. **K. Burrage and J. C. Butcher:** *"Non-linear stability of a general class of differential equation methods"*. BIT, **20** (1980), 185-203.
49. **K. Burrage, J. C. Butcher and F. N. Chipman:** *"An implementation of singly implicit Runge-Kutta methods"*. BIT, **20** (1980), 326-340.
50. **Z. Burnstein and A. A. Mirin:** *"Third order difference methods for hyperbolic equations"*. SIAM J. Appl. Math., **5** (1970), 547-557.
51. **J. C. Butcher:** *"Implicit Runge-Kutta processes"*. Math. Comput., **18** (1964), 55-64.
52. **J. C. Butcher:** *"A stability property of implicit Runge-Kutta methods"*. BIT, **15** (1975), 358-361.
53. **J. C. Butcher:** *"On the implementation of implicit Runge-Kutta methods"*. BIT, **16** (1976), 237-240.
54. **J. C. Butcher:** *"A transformed implicit Runge-Kutta method"*. J. Assoc. Comput. Mach., **26** (1979), 731-738.

55. **J. C. Butcher:** *"The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods"*. Wiley, New York, 1987.
56. **J. C. Butcher, K. Burrage and F. N. Chipman:** *"STRIDE: Stable Runge-Kutte integrator for differential equations"*. Report Series No. 150 (Computational Mathematics No. 20). Department of Mathematics, University of Auckland, Auckland, New Zealand, 1979.
57. **J. C. Butcher and J. R. Cash:** *"Towards efficient Runge-Kutta methods for stiff systems"*. SIAM J. Numer. Anal., **27** (1990), 753-761.
58. **J. G. Calvert and J. N. Pitts:** *"Photochemistry"*. Wiley, New York, 1966.
59. **M. Calvo, F. Lisbona and J. Montijano:** *"On the stability of variable-stepsize Nordsieck BDF methods"*. SIAM J. Numer. Anal., **24** (1987), 844-854.
60. **G. R. Carmichael and L. K. Peters:** *"An Eulerian transport/transformation/removal model for SO₂ and sulphate: I. Model development"*. Atmos. Environ., **18** (1984), 937-952.
61. **J. R. Cash:** *"On a class of cyclic methods for the numerical solution of stiff systems of O.D.E.'s"*. BIT, **17** (1977), 270-280.
62. **J. R. Cash:** *"Stable recursions with applications to the numerical solution of stiff systems"*. Academic Press, London, 1979.
63. **W. L. Chameides and D. H. Stedman:** *"Ozone formation from NO_x in 'Clear Air'"*. Environ. Sci. Tech., **10** (1979), 150-153.
64. **T. F. Chan and K. R. Jackson:** *"Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms"*. SIAM J. Sci. Statist. Comput., **5** (1984), 533-542.
65. **T. F. Chan and K. R. Jackson:** *"The use of iterative linear-equation solvers in codes for large systems of stiff IVP's for ODE's"*. SIAM J. Sci. Statist. Comput., **7** (1986), 378-417.
66. **J. S. Chang, R. A. Brost, I. S. A. Isaksen, S. Madronich, P. Middleton, W. R. Stockwell and C. J. Walcek:** *"A three dimensional Eulerian acid deposition model: Physical concepts and formulation"*. J. Geophys. Res., **92** (1987), 14691-14700.
67. **D. P. Chock:** *"A comparison of numerical methods for solving advection equations -II"*. Atmos. Environ., **19** (1985), 571-586.
68. **D. P. Chock:** *"A comparison of numerical methods for solving advection equations -III"*. Atmos. Environ., **25A** (1991), 853-871.
69. **D. P. Chock and A. M. Dunker:** *"A comparison of numerical methods for solving advection equations"*. J. Comp. Phys., **31** (1983), 352-362.

70. **J. Christensen:** *"Testing advection schemes in a three dimensional air pollution model"*. Math. Comput. Modelling, **18** (1993), 75-88.
71. **O. Christensen and L. P. Prahm:** *"A pseudospectral model for dispersion of atmospheric pollutants"*. J. Appl. Meteor. **15** (1976), 1284-1294.
72. **A. T. Chronopoulos and C. T. Pedro:** *"Iterative methods for nonsymmetric systems in DAE's and stiff ODE's codes"*. Mathematics and Computers in Simulation, **35** (1993), 1-22.
73. **T. L. Clark:** *"Numerical simulations with a three-dimensional cloud model: Lateral boundary condition experiments and multicellular severe storm simulations"*. J. Atmos. Sci., **36** (1979), 2191-2215.
74. **T. L. Clark and W. D. Hall:** *"A numerical experiment on stochastic condensation theory"*. J. Atmos. Sci., **36** (1979), 470-483.
75. **J. W. Cooley and J. W. Tukey:** *"An algorithm for the machine calculation of complex Fourier series"*. Math. Comp., **19** (1965), 297-301.
76. **R. Courant, K. O. Friedrichs and H. Lewy:** *"Über die partiellen Differenzengleichungen der mathematischen Physik"*. Math. Ann., **100** (1928), 32-74.
77. **J. Crank and P. Nicholson:** *"A practical method for numerical integration of solutions of partial differential equations of heat conduction type"*. Proc. Cambridge Philos. Soc., **43** (1947), 50-67.
78. **M. Crouzeix:** *"Sur l'approximation des equations differentielles operationelles lineaires par des methodes de Runge-Kutta"*. Ph D Thesis, L'Universite Paris VI, Paris, 1975.
79. **M. Crouzeix:** *"Sur la B-stabilite des methodes de Runge-Kutta"*. Numer. Math., **32** (1979), 75-82.
80. **M. Crouzeix and F. J. Lisbona:** *"The convergence of variable stepsize variable formula methods"*. SIAM J. Numer. Anal., **21** (1984), 397-415.
81. **W. P. Crowley:** *"Numerical advection experiments"*. Mon. Weath. Rev., **96** (1968), 1-11.
82. **C. W. Cryer:** *"A new class of highly stable methods: A_0 -stable methods"*. BIT, **13** (1973), 153-159.
83. **A. R. Curtis:** *"Jacobian matrix properties and their impact on choice of software for stiff ODE problems"*. I. M. A. J. Numer. Anal., **3** (1983), 397-415.
84. **A. R. Curtis:** *"Stiff ODE initial value problems and their solution"*. Report No CSS 182. Computer Science and Systems Division, Harwell Laboratory, Harwell, Oxfordshire, England 1985.

85. **G. Dahlquist:** *"Consistency and convergence in the numerical integration of ordinary differential equations"*. Math. Scand., **4** (1956), 33-53.
86. **G. Dahlquist:** *"Stability and error bounds in the numerical integration of ordinary differential equations"*. Trans. Roy. Inst. Tech. 130, Stockholm, 1959.
87. **G. Dahlquist:** *"A special stability problem for linear multistep methods"*. BIT, **3** (1963), 27-43.
88. **G. Dahlquist:** *"The sets of smooth solutions of differential and difference equations"*. In: **"STIFF DIFFERENTIAL SYSTEMS"** (R. A. Willoughby, ed.), pp. 60-74. Lecture Notes in Mathematics, No. 506, Plenum Press, New York, 1974.
89. **G. Dahlquist:** *"Error analysis for a class of methods for stiff nonlinear initial value problems"*. In: **"NUMERICAL ANALYSIS"**, Proceedings, Dundee 1975 (G. A. Watson, ed.), pp. 60-74. Lecture Notes in Mathematics, No. 506, Springer, Berlin, 1976.
90. **G. Dahlquist:** *"On the relation of G -stability to other concepts for linear multistep methods"*. In: **"TOPICS IN NUMERICAL ANALYSIS III"** (J. H. Miller, ed.), pp. 67-80. Academic Press, London, 1977.
91. **G. Dahlquist:** *"Some properties of linear multistep and one leg methods for ordinary differential equations"*. In: **"WORKING PAPERS FOR THE 1979 SIGNUM MEETING ON NUMERICAL ORDINARY DIFFERENTIAL EQUATIONS"** (R. D. Skeel, ed.), pp. 1.1-1.3. Department of Computer Science, University of Illinois at Urbana-Champaign, Illinois, 1979.
92. **G. Dahlquist:** *"On transformation of graded matrices with application to stiff ODE's"*. Numer. Math., **47** (1985), 363-385.
93. **G. Dahlquist and Å. Björck:** *"Numerical methods"*. Prentice-Hall, Englewood Cliffs, N. J., 1974.
94. **G. Dahlquist, W. Liniger and O. Nevanlinna:** *"Stability of two-step methods for variable integration steps"*. SIAM J. Numer. Anal., **20** (1983), 1071-1085.
95. **G. Dahlquist and G. Söderlind:** *"Some problems related to stiff nonlinear differential systems"*. In: **"COMPUTING METHODS IN APPLIED SCIENCES AND ENGINEERING"**, Vol. V (R. Glowinski and J. L. Lions, eds.), pp. 57-84. North-Holland, Amsterdam, 1984.
96. **C. Daly and J. J. Du Croz:** *"Performance of a subroutine library on vector-processing machines"*. Computer Physics Communications, **37** (1985), 181-186.
97. **J. W. Daniel and R. E. Moore:** *"Computation and theory in ordinary differential equations"*. Freeman, San Francisco, California, 1970.

98. **I. Daubechies:** *"Ten lectures on wavelets"*. CBMS-NSF Regional Conference Series in Applied Mathematics, Society of Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.
99. **K. Dekker and J. G. Verwer:** *"Stability of Runge-Kutta methods for stiff non-linear equations"*. North-Holland, Amsterdam, 1984.
100. **F. J. Dentener and P. J. Crutzen:** *"Reaction of N_2O_2 on tropospheric Aerosols: Impact on the global distribution of NO_x , O_3 and OH "*. J. Geophys. Res. **98** (1993), 7149-7163.
101. **R. Derwent:** *"Evaluation of a number of chemical mechanisms for their application in models describing the formation of ozone in Europe"*. Atmos. Environ. **24A** (1990), 2615-2624.
102. **R. Derwent:** *"Evaluation of the chemical mechanism employed in the EMEP photochemical oxidant model"*. Atmos. Environ. **27A** (1993), 277-279.
103. **R. Derwent, P. Grennfelt and Ø. Hov:** *"Photochemical oxidants in the atmosphere"*. Status report prepared for the Nordic Council of Ministers; No. L90/82. Swedish Environmental Research Insitute, Box 47086, 402 48 Gothenburg, Sweden, 1990.
104. **R. C. Derwent, P. Grennfelt and Ø. Hov:** *"Second generation abatement strategies for NO_x , NH_3 , SO_2 and VOC"*. Paper draft, 1992.
105. **P. Deuflhard:** *"Order and stepsize control in extrapolation methods"*. Numer. Math., **41** (1983), 399-422.
106. **P. Deuflhard:** *"Recent progress in extrapolation methods for ordinary differential equations"*. SIAM Rev., **27** (1985), 505-535.
107. **P. Deuflhard, E. Hairer and M. Zugh:** *"One step and extrapolation methods for differential-algebraic equations"*. Numer. Math., **51** (1987), 501-516.
108. **P. Deuflhard and U. Nowak:** *"Efficient numerical simulation and identification of Large Reaction Systems"*. Ber. Bensendes Phys. Chem., **90** (1986), 940-946.
109. **P. Deuflhard, U. Nowak and M. Wulkow:** *"Recent development in chemical computing"*. Computers Chem. Engng., **14** (1990), 1249-1258.
110. **H.-J. Diekhoff, P. Lory, H. J. Oberle, H. Y. J. Pesh, P. Rentrop and R. Seydel:** *"Comparing routines for the numerical solution of initial value problems of ordinary differential equations in multiple shooting"*. Numer. Math., **27** (1977), 449-469.
111. **B. Dimitriadis, M. C. Dodge, J. J. Bufalini, K. L. Demerjian and A. P. Altshuler:** *"Comment on 'Ozone formation from NO_x in Clear Air' by Chaimedes and Stedman"*. Environ. Sci. Tech., **10** (1976), 934-936.

112. **J. J. Dongarra, J. R. Bunch, C. B. Moler and G. W. Stewart:** *"LINPACK: Users guide"*. SIAM Publications, Philadelphia, 1979.
113. **J. J. Dongarra, J. J. Du Croz, S. Hammarling and R. J. Hanson:** *"A proposal for an extended set of Fortran basic linear algebra subprograms"*. ACM SIGNUM Newsletter, **20** (1985), 1-18.
114. **J. J. Dongarra and S. C. Eisenstat:** *"Squeezing the most out of an algorithm in Cray Fortran"*. ACM Trans. Math. Software, **10** (1984), 219-230.
115. **J. J. Dongarra, F. G. Gustavson and A. Karp:** *"Implementing linear algebra algorithms for dense matrices on a vector pipeline machine"*. SIAM Rev., **26** (1984), 91-112.
116. **J. J. Dongarra, G. K. Leaf and M. Minkoff:** *"A preconditioned conjugate gradient method for solving a class of non-symmetric linear systems"*. Report No. ANL-81-71. Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois, USA, 1986.
117. **J. J. Dongarra and A. H. Sameh:** *"On some parallel banded system solvers"*. Parallel Computing, **1** (1984), 223-235.
118. **J. R. Dormant and P. J. Prince:** *"New Runge-Kutta algorithms for numerical simulation in dynamical astronomy"*. Celestial Mechanics, **18** (1978), 223-232.
119. **J. R. Dormant and P. J. Prince:** *"A family of embedded Runge-Kutta formulae"*. J. Comput. Appl. Math., **6** (1980), 19-26.
120. **H. Dovland and J. Saltbones:** *"Emissions of sulphur dioxide in Europe in 1978"*. EMEP/CCC - Report 2/79. Norwegian Institute for Air Research (NILU), P. O. Box 130, N-2001 Lillestrøm, Norway, 1979.
121. **P. F. Dubois, A. Greenbaum and G. H. Rodrigue:** *"Approximating the inverse of a matrix for the use of iterative algorithms on vector processors"*. Computing, **22** (1979), 308-323.
122. **J. J. Du Croz and J. Wasniewski:** *"Basic linear algebra computations on the Sperry ISP"*. Supercomputer, **21** (1987), 45-54.
123. **I. S. Duff:** *"Pivot selection and row ordering in Givens reduction of sparse matrices"*. Computing, **13** (1974), 239-248.
124. **I. S. Duff:** *"MA28: a set of FORTRAN subroutines for sparse unsymmetric linear equations"*. Report No. R8730. Computer Science and Systems Division, A.E.R.E., Harwell, England, 1977.

125. **I. S. Duff:** *"Practical comparisons of codes for the solution of sparse linear systems"*. In: **"SPARSE MATRIX PROCEEDINGS 1978"** (I. S. Duff and G. W. Stewart, eds.), pp. 107-134. Society for Industrial and Applied Mathematics, Philadelphia, 1979.
126. **I. S. Duff:** *"The solution of sparse linear equations on the CRAY1"*. Report No. CSS 125 (Revised). Computer Science and Systems Division, A.E.R.E., Harwell, Oxfordshire, England, 1983.
127. **I. S. Duff:** *"Data structures, algorithms and software for sparse matrices"*. In: **"SPARSITY AND ITS APPLICATIONS"** (D. J. Evans, ed.), pp. 1-29. Cambridge University Press, Cambridge-London, 1985.
128. **I. S. Duff:** *"The solution of large-scale least-squares problems on supercomputers"*. Report No. CSS 225. Computer Science and Systems Division, A.E.R.E., Harwell, Oxfordshire, England, 1988.
129. **I. S. Duff:** *"Experience with the HARWELL Subroutine Library on the CRAY-2"*. Report No. CSS 227. Computer Science and Systems Division, A.E.R.E., Harwell, England, 1988.
130. **I. S. Duff, A. M. Erisman and J. K. Reid:** *"Direct methods for sparse matrices"*. Oxford University Press, Oxford-London, 1986.
131. **I. S. Duff, N. I. M. Gould, M. Lescrenier and J. K. Reid:** *"The multifrontal method in a parallel environment"*. Report No. CSS 211. Computer Science and Systems Division, A.E.R.E., Harwell, England, 1987.
132. **I. S. Duff, R. G. Grimes and J. C. Lewis:** *"Sparse matrix test problems"*. ACM Trans. Math. Software, **15** (1989), 1-14.
133. **I. S. Duff, R. G. Grimes, J. C. Lewis and G. W. Poole, Jr.:** *"Sparse matrix test problems"*. SIGNUM Newsletter, **17**, 2 (1989), 22.
134. **I. S. Duff and J. K. Reid:** *"A comparison of sparsity orderings for obtaining a pivotal sequence in Gaussian elimination"*. J. Inst. Math. Applics., **14** (1974), 281-291.
135. **I. S. Duff and J. K. Reid:** *"On the reduction of sparse matrices to condensed form by similarity transformations"*. J. Inst. Math. Applics., **15** (1975), 217-224.
136. **I. S. Duff and J. K. Reid:** *"A comparison of some methods for the solution of sparse overdetermined systems of linear equations"*. J. Inst. Math. Applics., **17** (1976), 267-280.
137. **I. S. Duff and J. K. Reid:** *"Performance evaluation of codes for sparse matrices"*. Report No. CSS 66. Computer Science and Systems Division, A.E.R.E., Harwell, England, 1978.

138. **I. S. Duff and J. K. Reid:** *"Some design features of a sparse matrix code"*. ACM Trans. Math. Software, **5** (1979), 18-35.
139. **I. S. Duff and J. K. Reid:** *"The multifrontal solution of indefinite sparse symmetric linear systems"*. Report No. CSS 122. Computer Science and Systems Division, A.E.R.E., Harwell, England, 1982.
140. **I. S. Duff, J. K. Reid and J. A. Scott:** *"The use of profile reduction algorithms with a frontal code"*. Report No. CSS 224. Computer Science and Systems Division, A.E.R.E., Harwell, England, 1988.
141. **E. G. D'yakonov:** *"On certain difference schemes with split-up operators for multi-dimensional stationary problems"*. Zh. Vychisl. Mat. Mat. Fiz., **2** (1962), 57-79.
142. **E. G. D'yakonov:** *"Difference schemes of second-order accuracy with a splitting operator for parabolic equations without mixed derivatives"*. Zh. Vychisl. Mat. Mat. Fiz., **4** (1964), 935-943.
143. **R. C. Easter:** *"Two modified versions of Bott's positive definite numerical advection scheme"*. Report No. PNL-SA-19677. Pacific Northwest Laboratory, Richland, Washington 99352, USA, 1991.
144. **A. Ebel, H. Hass, H. J. Jakobs, M. Laube and M. Memmesheimer:** *"Transport of atmospheric minor constituents as simulated by the EURAD model"*. In: **"PROCEEDING OF THE EMEP WORKSHOP ON PHOTOOXIDANT MODELLING FOR LONG-RANGE TRANSPORT IN RELATION TO ABATEMENT STRATEGIES"** (J. Pankrath, ed.), pp. 142-154. Umweltsamt, Berlin, Germany, 1991.
145. **B. L. Ehle:** *"High order A-stable methods for the numerical solution of ODE's"*. BIT, **8** (1968), 276-278.
146. **B. L. Ehle:** *"On Pade approximation to the exponential function and A-stable methods for the numerical solution of initial value problems"*. Report No. CSRR 2010. Computer Science Department, University of Waterloo, Waterloo, Ontario, Canada, 1969.
147. **S. C. Eisenstat:** *"Efficient implementation of a class of preconditioned conjugate gradient methods"*. SIAM J. Sci. Statist. Comput., **2** (1981), 1-4.
148. **S. C. Eisenstat, H. C. Elman and M. H. Schultz:** *"Variational methods for nonsymmetric systems of linear equations"*. SIAM J. Numer. Anal., **20** (1983), 345-357.
149. **S. C. Eisenstat, M. C. Gursky, M. H. Schultz and A. H. Sherman:** *"The YALE sparse matrix package: The symmetric codes"*. Internat. J. Numer. Meth. Engng., **18** (1982), 1145-1151.

150. **S. C. Eisenstat, M. C. Gursky, M. H. Schultz and A. H. Sherman:** *"The YALE sparse matrix package: The non-symmetric codes"*. Report No. 114. Department of Computer Science, Yale University, New Haven, Connecticut, 1977.
151. **S. C. Eisenstat, M. H. Schultz and A. H. Sherman:** *"Software for sparse Gaussian elimination with limited core storage"*. In: **"SPARSE MATRIX PROCEEDINGS 1978"** (I.S.Duff and G. W. Stewart, eds.), 135-153. Society for Industrial and Applied Mathematics, Philadelphia, 1979.
152. **S. C. Eisenstat, M. H. Schultz and A. H. Sherman:** *"Algorithms and data structure for sparse symmetric Gaussian elimination"*. SIAM J. Sci. Statist. Comput., **2** (1981), 225-237.
153. **A. Eliassen, Ø. Hov, I. S. Isaksen, J. Saltbones and F. Stordal:** *"A Lagrangian long-range model with atmospheric boundary layer chemistry"*. J. App. Meteor., **21** (1982), 1645-1661.
154. **A. Eliassen and J. Saltbones:** *"Modelling of long-range transport of sulphur over Europe: a two-year model run and some model experiments"*. Atmos. Environ., **17** (1983), 1457-1473.
155. **W. H. Enright:** *"Second order derivative methods for stiff ordinary differential equations"*. SIAM J. Numer. Anal., **11** (1974), 321-331.
156. **W. H. Enright and T. E. Hull:** *"Test results on initial value methods for non-stiff ordinary differential equations"*. SIAM J. Numer. Anal., **13** (1976), 944-961.
157. **W. H. Enright, T. E. Hull and B. Lindberg:** *"Comparing methods for stiff systems of ODE's"*. BIT, **15** (1975), 10-48.
158. **W. H. Enright, K. R. Jackson, S. R. Nørsett and P. G. Thomsen:** *"Effective solution of discontinuous IVPs using a Runge-Kutta pair with interpolants"*. Numerical Analysis Report No. 113. Department of Mathematics, University of Manchester, Manchester, England 1986.
159. **W. H. Enright and M. S. Kamel:** *"Automatic partitioning of stiff systems and exploiting the resulting structures"*. ACM Trans. Math. Software, **5** (1979), 374-385.
160. **D. J. Evans:** *"The use of preconditioning in iterative methods for solving linear equations with symmetric positive definite matrices"*. J. Inst. Math. Applics., **4** (1968), 295-314.
161. **D. J. Evans:** *"The analysis and application of sparse matrix algorithms in the finite element method"*. In: **"THE MATHEMATICS OF FINITE ELEMENTS AND APPLICATIONS"** (J. R. Whiteman, ed.), pp. 427-447. Academic Press, London, 1973.

162. **D. J. Evans:** *"Iterative sparse matrix algorithms"*. In: **"SOFTWARE FOR NUMERICAL MATHEMATICS"** (D. J. Evans, ed.), pp. 49-83. Academic Press, London, 1974.
163. **D. J. Evans:** *"Iterative methods for sparse matrices"*. In: **"SPARSITY AND ITS APPLICATIONS"** (D. J. Evans, ed.), pp. 45-111. Cambridge University Press, Cambridge-London, 1985.
164. **D. K. Fadееv and V. N. Fadееva:** *"Computational methods of linear algebra"*. Freeman, San Francisco, 1963.
165. **L. A. Farrow and D. Edelson:** *"The steady-state approximation: fact or fiction"*. Int. J. Chem. Kinet., **6** (1974), 787-800.
166. **S. O. Fatunla:** *"Numerical methods for initial value problems in ordinary differential equations"*. Academic Press, London, 1988.
167. **E. Fehlberg:** *"Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme"*. Computing, **6** (1970), 61-71.
168. **C. K. Forester:** *"Higher order monotonic convective difference schemes"*. J. Comput. Phys., **23** (1977), 1-22.
169. **B. Fornberg:** *"On a Fourier method for the integration of hyperbolic equations"*. SIAM J. Numer. Anal., **12** (1975), 509-528.
170. **G. E. Forsythe, M. A. Malcolm and C. B. Moler:** *"Computer methods for mathematical computations"*. Prentice-Hall, Englewood Cliffs, N. J., 1977.
171. **G. E. Forsythe and C. B. Moler:** *"Computer solution of linear algebraic equations"*. Prentice-Hall, Englewood Cliffs, N. J., 1967.
172. **D. G. Fox and S. A. Orszag:** *"Pseudospectral approximation to two-dimensional turbulence"*. J. Comput. Phys., **11** (1973), 612-619.
173. **R. Frank J. Schneid and C. W. Ueberhuber:** *"The concept of B-convergence"*. SIAM J. Numer. Anal., **18** (1981), 753-780.
174. **R. Frank and C. W. Ueberhuber:** *"Iterated defect correction for differential equations. Part I: Theoretical results"*. Computing, **20** (1978), 207-228.
175. **R. Freund:** *"On conjugate gradient type methods and polynomial preconditioners for a class of complex non-Hermitian matrices"*. Numer. Math., **57** (1990), 285-312.
176. **R. F. Freund and D. S. Conwell:** *"Superconcurrency: A form of distributed heterogeneous computing"*. Supercomputing Rev., **Vol. 3**, No. 10, 1990, 47-50.

177. K. A. Gallivan, M. T. Heath, E. Ng, J. M. Ortega, B. W. Peyton, R. J. Plemmons, C. H. Romine, A. H. Sameh and R. G. Voigt: *"Parallel algorithms for matrix computations"*. Society for Industrial and Applied Mathematics, Philadelphia, 1990.
178. K. A. Gallivan, W. Jalby and U. Meier: *"The use of BLAS3 in linear algebra on a parallel processor with hierarchical memory"*. SIAM J. Sci. Statist. Comput., **8** (1987), 1079-1084.
179. K. A. Gallivan, R. J. Plemmons and A. H. Sameh: *"Parallel algorithms for dense linear algebra computations"*. SIAM Rev., **32** (1990), 54-135.
180. F. R. Gantmacher: *"Theory of matrices"*. Vol. 1 and Vol. 2. Chelsea, New York, 1959.
181. J. M. Gary: *"A comparison of two difference schemes used for numerical weather prediction"*. J. Comput. Phys., **4** (1969), 279-305.
182. C. W. Gear: *"Numerical initial value problem for ordinary differential equations"*. Prentice-Hall, Englewood Cliffs, N. J., USA, 1971.
183. C. W. Gear: *"The automatic integration of ordinary differential equations"*. Comm. ACM, **14** (1971), 176-179.
184. C. W. Gear: *"Algorithm 407, DIFSUB for solution of ordinary differential equations"*. Comm. ACM, **14** (1978), 185-190.
185. C. W. Gear: *"Numerical error in sparse linear equations"*. Report No. UIUCDCS-F-75-885. Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1975.
186. C. W. Gear: *"Stability of variable step methods for ordinary differential equations"*. Report UIUCDCS-R-78-938. Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1978.
187. C. W. Gear: *"Runge-Kutta starters for multistep methods"*. ACM Trans. Math. Software, **6** (1980), 263-279.
188. C. W. Gear: *"Numerical solution of ordinary differential equations: is there anything left to do?"*. SIAM Review, **23** (1981), 10-24.
189. C. W. Gear and Y. Saad: *"Iterative solution of linear equations in ODE codes"*. SIAM J. Sci. Statist. Comput., **4** (1983), 583-601.
190. C. W. Gear and K. W. Tu: *"The effect of variable mesh on the stability of multistep methods"*. SIAM J. Numer. Anal., **11** (1974), 1024-1043.
191. C. W. Gear and D. S. Watanabe: *"Stability and convergence of variable multistep methods"*. SIAM J. Numer. Anal., **11** (1974), 1044-1053.

192. J. A. George and M. T. Heath: "*Solution of sparse linear least squares problems using Givens rotations*". *Lin. Alg. Appl.*, **34** (1980), 69-83.
193. J. A. George, M. T. Heath and E. Ng: "*A comparison of some methods for solving sparse linear least squares problems*". *SIAM J. Sci. Statist. Comput.*, **4** (1983), 177-187.
194. J. A. George, M. T. Heath and R. J. Plemmons: "*Solution of large-scale linear least squares problems using auxiliary storage*". *SIAM J. Sci. Statist. Comput.*, **2** (1981), 416-429.
195. J. A. George and J. W. Liu: "*Computer solution of large sparse positive definite systems*". Prentice-Hall, Englewood Cliffs, N. J., 1981.
196. J. A. George and J. W. Liu: "*Householder reflections versus Givens rotations in sparse orthogonal decomposition*". *Lin. Alg. Appl.*, **88** (1987), 223-238.
197. J. A. George, J. W. Liu and E. Ng: "*User guide for SPARSPACK: Waterloo sparse linear equations package*". Report No. CS-78-30. Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1980.
198. J. A. George, J. W. Liu and E. Ng: "*Row ordering schemes for sparse Givens rotations: I. Bipartite graph model*". *Lin. Alg. Appl.*, **61** (1984), 55-81.
199. J. A. George, J. W. Liu and E. Ng: "*Row ordering schemes for sparse Givens rotations: II. Implicit graph model*". *Lin. Alg. Appl.*, **75** (1986), 203-223.
200. J. A. George, J. W. Liu and E. Ng: "*Row ordering schemes for sparse Givens rotations: III. Analyses for a model problem*". *Lin. Alg. Appl.*, **75** (1986), 225-240.
201. J. A. George, J. W. Liu and E. Ng: "*A data structure for sparse QR and LU factors*". Report No. CS 85-16. Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1985.
202. J. A. George and E. Ng: "*An implementation of Gaussian elimination with partial pivoting for sparse systems*". *SIAM J. Sci. Statist. Comput.*, **6** (1985), 390-405.
203. J. A. George and E. Ng: "*Symbolic factorization for sparse Gaussian elimination with partial pivoting*". *SIAM J. Sci. Statist. Comput.*, **8** (1987), 877-898.
204. M. W. Gery, G. Z. Whitten, J. P. Killus and M. C. Dodge: "*A photochemical kinetics mechanism for urban and regional computer modeling*". *J. Geophys. Res.*, **94** (1989), 12925-12956.
205. F. A. Gifford: "*Horizontal diffusion in the atmosphere: A Lagrangian dynamical theory*". *Atmos. Environ.*, **16** (1982), 505-512.

206. **J. R. Gilbert:** *"An efficient parallel sparse partial pivoting algorithm"*. Report No. 88/45052-1. Chr. Michelsen Institute, Department of Science and Technology, Centre for Computer Science, Fantoftvegen 38, N-5036 Fantoft, Bergen, Norway, 1988.
207. **J. R. Gilbert and T. Peierls:** *"Sparse partial pivoting in time proportional to arithmetic operations"*. SIAM J. Sci. Statist. Comput., **9** (1988), 862-874.
208. **Ph. E. Gill, W. Murray, M. A. Saunders and M. H. Wright:** *"Maintaining LU factors of a general sparse matrix"*. Lin. Alg. Appl., **88** (1987), 239-270.
209. **I. Gladwell:** *"The NAG library Runge-Kutta subroutines"*. ACM Trans. Math. Software, **5** (1979), 386-400.
210. **G. H. Golub and C. F. Van Loan:** *"Matrix computation"*. J. Hopkins University Press, Baltimore, Maryland, 1983.
211. **D. Gotlieb and S. A. Orszag:** *"Mathematical analysis of spectral methods: Theory and application"*. NSF-CBMS Monography No. 25, SIAM, Philadelphia, 1977.
212. **B. W. Gragg:** *"On extrapolation algorithms for ordinary initial value problems"*. SIAM J. Numer. Anal., Ser. B, **2** (1965), 384-403.
213. **A. Granneltvedt:** *"A survey of finite-difference schemes for the primitive equations for a barotropic fluid"*. Monthly Weather Review, **97** (1969), 384-404.
214. **N. Grandjean:** *"The modified equation approach to flux-corrected transport"*. J. Comput. Phys., **91** (1990), 424-440.
215. **P. Grennfelt, A. Eliassen, Ø. Hov, R. Berkowicz and G. Nordlund:** *"Atmospheric chemistry, transport and deposition of nitrogen oxides"*. Status report prepared for the working group on nitrogen oxides within ECE's "Convention on Long-Range Transboundary Air Pollution". Nordisk Ministerråd, Store Strandstræde 18, DK-1255 Copenhagen K, Denmark, 1986.
216. **P. Grennfelt, J. Saltbones and J. Schjoldager:** *"Oxidant data collection in 1985-87 (OXIDATE)"*. Report No. NILU 22/87. Norwegian Institute for Air Pollution, Postboks 64, N-2001 Lillestrøm, Norway, 1987.
217. **Ph. M. Gresho and R. L. Lee:** *"Don't suppress the wiggles- they're telling you something"*. Computers and Fluids, **9** (1981), 223-253.
218. **R. D. Grigorieff:** *"Stability of multistep methods on variable grids"*. Numer. Math., **42** (1983), 359-377.
219. **R. D. Grigorieff and P. J. Paes-Leme:** *"On the zero-stability of the 3-step BDF-formula on nonuniform grids"*. BIT, **24** (1984), 85-91.

220. **G. K. Gupta:** *"A note about overhead costs in ODE solvers"*. ACM Trans Math. Software, **6** (1981), 319-326.
221. **G. K. Gupta, R. Sacks-Davis and P. E. Tescher:** *"A review of recent developments in solving ODE's"*. ACM Comput. Surv., **17** (1985), 5-47.
222. **F. G. Gustavson:** *"Some basic techniques for solving sparse systems of linear equations"*. In: **"SPARSE MATRICES AND THEIR APPLICATIONS"** (D. J. Rose and R. A. Willoughby, eds.), 41-52. Plenum Press, New York, 1972.
223. **F. G. Gustavson:** *"Two fast algorithms for sparse matrices: multiplication and permuted transposition"*. ACM Trans. Math. Software, **4** (1978), 250-269.
224. **A. L. Hageman and D. M. Young:** *"Applied iterative methods"*. Academic Press, New York, 1981.
225. **E. Hairer and Ch. Lubich:** *"Extrapolation at stiff differential equations"*. Report. Dept. de mathematiques, Universite de Geneve. Case postale 240, CH-1211 Geneve 24, Switzerland, 1987.
226. **E. Hairer, S. P. Nørsett and G. Wanner:** *"Solving ordinary differential equations: I. Nonstiff problems"*. Springer, Berlin, 1987.
227. **E. Hairer and G. Wanner:** *"Solving ordinary differential equations: II. Stiff problems"*. Springer, Berlin, 1991.
228. **R. M. Harrison, Z. Zlatev and C. J. Ottley:** *"A comparison of the predictions of an Eulerian atmospheric transport-chemistry model with experimental measurements over the North Sea"*. Atmos. Environ., **28** (1992), 497-516.
229. **Harwell Subroutine Library:** *"User's Guide"*. A.E.R.E, Harwell, England, 1983.
230. **H. Hass, M. Memmesheimer, H. Geiß, H. J. Jacobs, M. Laube and A. Ebel:** *"Simulation of the Chernibyl radioactive cloud over Europe using the EURAD model"*. Atmospheric Environment, **24A** (1990), 673-692.
231. **H. Hass, A. Ebel, H. Feldmann, H. J. Jacobs and M. Memmesheimer:** *"Evaluation studies with a regional chemical transport model (EURAD) using air quality data from the EMEP monitoring network"*. Atmospheric Environment, **27A** (1993), 867-887.
232. **N. Z. Heidam:** *"Nitrogen depositions to the Baltic Sea: Experimental and model estimations"*. Atmospheric Environment, **27A** (1993), 815-822.
233. **P. Henrici:** *"Discrete variable methods in ordinary differential equations"*. Wiley, New York, 1963.

234. **O. Hertel, R. Berkowicz, J. Christensen and Ø. Hov:** *"Test of two numerical schemes for use in atmospheric transport-chemistry models"*. *Atmospheric Environment*, **27A** (1993), 2591-2611.
235. **E. Hesstvedt, Ø. Hov and I. A. Isaksen:** *"Quasi-steady-state approximations in air pollution modelling: comparison of two numerical schemes for oxidant prediction"*. *International Journal of Chemical Kinetics*, **10** (1978), 971-994.
236. **M. R. Hestenes and E. Stiefel:** *"Methods of conjugate gradients for solving linear systems"*. *J. Res. Nat. Bur. Stand.*, **49** (1952), 409-436.
237. **B. B. Hicks and D. R. Matt:** *"Combining biology, chemistry and meteorology in modelling and measuring dry deposition"*. *J. Atmos. Chem.*, **6** (1988), 117-133.
238. **N. J. Higham:** *"A survey of condition number estimation for triangular matrices"*. *SIAM Rev.*, **29** (1987), 575-596.
239. **N. J. Higham:** *"Fortran codes for estimating the one-norm of a real or complex matrix with application to condition estimation"*. Numerical Analysis Report No. 135. Department of Mathematics, University of Manchester, Manchester M13 9PL, ENGLAND, 1987.
240. **A. C. Hindmarsh:** *"GEAR: ordinary differential equation solver"*. Report No. UCRL-51186. Lawrence Livermore Laboratory, Livermore, California, USA, 1971.
241. **A. C. Hindmarsh:** *"LSODE and LSODI, two new solvers of initial value ordinary differential equations"*. *ACM SIGNUM Newsletter*, **15** (1980), 10-11.
242. **A. C. Hindmarsh:** *"ODEPACK, a systemized collection of ODE solvers"*. In: **"IMACS TRANSACTIONS ON SCIENTIFIC COMPUTATION"**, Vol. 1 (R. S. Stepleman, ed.). North-Holland, Amsterdam, 1983.
243. **E. Holm and G. Erbes:** *"Solving the advection equation in flows with sharp gradients using Bott's flux scheme"*. Report No. DM-59. Department of Meteorology, Stockholm University, 1991.
244. **A. S. Householder:** *"Unitary triangularization of a nonsymmetric matrix"*. *J. Assoc. Comput. Mach.*, **5** (1958), 339-342.
245. **P. J. van der Houwen:** *"Construction of integration formulas for initial value problems"*. North-Holland, Amsterdam, 1977.
246. **Ø. Hov, Z. Zlatev, R. Berkowicz, A. Eliassen and L. P. Prahm:** *"Comparison of numerical techniques for use in air pollution models with nonlinear chemical reactions"*. *Atmospheric Environment*, **23** (1988), 967-983.

247. **T. E. Hull, W. H. Enright, B. M. Fellen and A. E. Sedgwick:** *"Comparing numerical methods for ordinary differential equations"*. SIAM J. Numer. Anal., **9** (1972), 603-637.
248. **A. S. Hunding:** *"Dissipative structures in reaction-diffusion systems: Numerical determination of bifurcations in the sphere"*. J. Chem. Phys., **72** (1980), 5241-5248.
249. **W. H. Hundsdorfer, B. Koren, M. van Loon and J. G. Verwer:** *"A positive finite difference advection scheme applied on locally refined grids"*. Report NM-R9309. Department of Numerical Mathematics, National Research Institute for Mathematics and Computer Science, P. O. Box 94079, Kruislaan 413, 1098 SJ Amsterdam, 1993.
250. **W. Hurewicz:** *"Lectures on ordinary differential equations"*. M.I.T. Press, Cambridge, Massachusetts, USA, 1958.
251. **T. Iversen:** *"Report of the activities at the Meteorological Synthesizing Centre - West during the fourth phase of EMEP, 1987-1989"*. Report 1/90, EMEP Meteorological Synthesizing Centre - West, Norwegian Meteorological Institute., P.O. Box 43, Blindern, N-0313 Oslo 3, NORWAY, 1990.
252. **T. Iversen, N. E. Halvorsen, S. Mylona and H. Sandnes:** *"Calculated budgets for airborne acidifying components in Europe, 1985, 1987, 1988 and 1990"*. Report No 1/91. EMEP Meteorological Synthesizing Centre - West, Norwegian Meteorological Institute, P.O. Box 43, Blindern, N-0313 Oslo 3, NORWAY, 1991.
253. **T. Iversen, N. E. Halvorsen, J. Saltbones and H. Sandnes:** *"Calculated budgets for airborne sulphur and nitrogen in Europe"*. Report No. 2/90. EMEP Meteorological Synthesizing Centre - West, The Norwegian Meteorological Institute, P. O. Box 43 - Blindern, N-0313 Oslo 3, NORWAY, 1990.
254. **T. Iversen, J. Saltbones, H. Sandnes, A. Eliassen and Ø. Hov:** *"Airborne transboundary transport of sulphur and nitrogen over Europe - Model description and calculations"*. Report No 2/89. EMEP Meteorological Synthesizing Centre - West, Norwegian Meteorological Institute, P.O. Box 43, Blindern, N-0313 Oslo 3, NORWAY, 1989.
255. **K. R. Jackson:** *"Variable stepsize variable order integrand approximation methods for the numerical solution of ordinary differential equations"*. Report No. 129-78. Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 1978.
256. **J. P. Jacobsen, H. K. Bildsøe and K. Schaumburg:** *"Application of density matrix formalism in NMR spectroscopy. II. The one-spin-1 case in anisotropic phase"*. J. Magn. Resonance, **23** (1976), 153-164.
257. **M. Jankowski and H. Wozniakowski:** *"Iterative refinement implies numerical stability"*. BIT, **17** (1977), 303-311.

258. **H. E. Jeffries and M. Saeger:** *"Comment on 'Ozone formation from NO_x in Clear Air' by Chaimedes and Stedman"*. Environ. Sci. Tech., **10** (1976), 936-937.
259. **R. Jeltsch and O. Nevanlinna:** *"Stability of explicit time discretizations for solving initial value problems"*. Numer. Math., **37** (1981), 61-91.
260. **R. Jeltsch and O. Nevanlinna:** *"Stability and accuracy of time-discretizations for initial value problems"*. Numer. Math., **40** (1982), 245-261.
261. **A. Jennings:** *"Matrix computations for engineers and scientists"*. Wiley, New York, 1977.
262. **P. Kaps:** *"Rosenbrock-type methods"*. In: **"NUMERICAL METHODS FOR SOLVING STIFF INITIAL VALUE PROBLEMS"**, Proceedings, Oberwolfach, 28.6.1981 Y 4.7.1981 (G. Dahlquist and R. Jeltsch, eds.). Department of Computer Science, University of Aachen, Aachen, Federal Republic of Germany, 1981.
263. **P. Kaps, S. W. H. Poon and T. D. Bui:** *"Rosenbrock methods for stiff ODE's: a comparison of Richardson extrapolation and embedding technique"*. Computing, **34** (1985), 17-40 .
264. **A. A. Khokhar, V. K. Prasanna, M. E. Shaaban and C.-L. Wang:** *"Heterogeneous computing: Challenges and opportunities"*. IEEE Computing, Vol **26**, No 6, 1993, 18-27 .
265. **D. Kincaid, T. Oppe, J. Respass and D. Young:** *"ITPACKV 2C: User's guide"*. Report No. CNA-191. Center for Numerical Analysis, University of Texas at Austin, Austin, Texas, 1984.
266. **D. Kincaid, J. Respass, D. Young and R. Grimes:** *"Algorithm 586, ITPACK 2C: A FORTRAN package for solving large sparse systems by adaptive accelerated iterative methods"*. ACM Trans. Math. Software, **8** (1982), 302-322.
267. **A. N. Konovalov:** *"The application of the splitting up method in dynamical problems in the theory of elasticity"*. Zh. Vychisl. Mat. Mat. Fiz., **4** (1964), 760-772.
268. **J. F. B. Kraaijevanger and J. Schneid:** *"On the solvability of the Runge-Kutta equations"*. Numer. Math., **59** (1991), 129-157.
269. **H. O. Kreiss:** *"Difference methods for stiff ordinary differential equations"*. SIAM J. Numer. Anal., **15** (1978), 21-58.
270. **H. O. Kreiss and J. Oliger:** *"Comparison of accurate methods for the integration of hyperbolic equations"*. Tellus, **24** (1972), 199-215.
271. **F. T. Krogh:** *"Predictor-corrector methods of high order with improved stability characteristics"*. J. Assoc. Comput. Mach., **14** (1966), 374-385.

272. **F. T. Krogh:** *"VODQ/SVDQ/DVDQ - variable order integrators for numerical solution of ordinary differential equations"*. Report. Jet Propulsion Laboratory, Pasadena, California, 1969.
273. **F. T. Krogh:** *"On testing a subroutines for the numerical solution of ordinary differential equations"*. J. Assoc. Comput. Mach., **20** (1973), 545-562.
274. **F. T. Krogh:** *"Algorithms for changing the step size"*. SIAM J. Numer. Anal., **10** (1973), 949-965.
275. **F. T. Krogh and K. Stewart:** *"Asymptotic ($h \rightarrow \infty$) absolute stability for BDF's applied to stiff equations"*. Report. Jet Propulsion Laboratory, Pasadena, California, 1981.
276. **F. T. Krogh and K. Stewart:** *"Implementation of variable step BDF methods for stiff ODE's"*. Report. Jet Propulsion Laboratory, Pasadena, California, 1981.
277. **D. J. Kuck, E. S. Davidson, D. H. Lawrie and A. H. Sameh:** *"Parallel surepcomputing today and the Cedar approach"*. Science, **231** (1986), 967-974.
278. **Y. Kurihara:** *"On the use of implicit and iterative methods for the time integration of the wave equation"*. Monthly Weather Review, **1** (1965), 33-46.
279. **B. Lamb, A. Guenter, D. Gay and H. Westberg:** *"A national inventory of biogenic hydrocarbon emissions"*. Atmos. Environ., **21** (1987), 1695-1705.
280. **J. D. Lambert:** *"Computational methods in ordinary differential equations"*. Academic Press, New York, 1973.
281. **J. D. Lambert:** *"Computational methods in ordinary differential equations"*. Academic Press, New York, 1992.
282. **P. Lancaster:** *"Theory of matrices"*. Academic Press, London, 1969.
283. **L. Lapidus and J. H. Seinfeld:** *"Numerical solution of ordinary differential equations"*. Academic Press, New York, 1971.
284. **C. Lawson, R. Hanson, D. Kincaid and F. Krogh:** *"Basic linear algebra subprograms for Fortran usage"*. ACM Trans. Math. Software, **5** (1979), 308-323.
285. **P. D. Lax:** *"Weak solutions of nonlinear hyperbolic equations and their numerical computation"*. Comm. Pure Appl. Math., **7** (1954), 159-193.
286. **P. D. Lax and B. Wendroff:** *"Systems of conservative laws"*. Comm. Pure Appl. Math., **13** (1960), 217-237.
287. **H. N. Lee:** *"An alternative pseudospectral model for pollutant transport, diffusion and deposition in the atmosphere"*. Atmospheric Environment, **15** (1981), 1017-1024.

288. **H. N. Lee:** *"A semi-Lagrangian transport scheme with spectral interpolation"*. J. Appl. Meteor., **35** (1993), 1908-1918.
289. **H. N. Lee and J. K. Shi:** *"Pseudospectral technique for calculation of atmospheric boundary layer flows"*. Journal of Climate and Applied Meteorology, **25** (1986), 1768-1774.
290. **C. Levit:** *"Grid communication on the Connection Machine: Analysis, performance and improvements"*. In: **"SCIENTIFIC APPLICATIONS OF THE CONNECTION MACHINE"** (H. D. Simon, ed.), pp. 316-322. World Scientific Publishing, Singapore, 1989.
291. **B. Lindberg:** *"Error estimation and iterative improvement for discretization algorithms"*. BIT, **20** (1980), 486-500.
292. **P. E. Long, Jr. and D. W. Pepper:** *"An examination of some simple numerical schemes for calculating scalar advection"*. J. Appl. Math., **20** (1981), 146-156.
293. **M. van Loon:** *"Testing interpolation and filtering techniques in connection with a semi-Lagrangian method"*. Atmos. Environ., **27A** (1993), 2351-2364.
294. **B. Lübker:** *"A model for estimating nitrogen oxide emissions in Europe"*. Working paper - WP-87-122. International Institute for Applied Systems Analysis (IIASA), A-2361 Laxenburg, Austria, 1987.
295. **B. Lübker and W. Schöpp:** *"A model to calculate natural VOC emissions from forests in Europe"*. Report No. WP-89-082. International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria, 1989.
296. **B. Lübker and S. Tilly:** *"The OECD-map emission inventory for SO₂, NO_x and VOC in Western Europe"*. Atmospheric Environment, **23** (1989), 3-15.
297. **G. Ludes and W. Ulrich:** *"Numerical simulation of plume-advection over a hill with coupled models"*. Meteor. Atmos. Phys., **39**(1988), 97-107.
298. **L. Lustman, B. Neta and W. Gragg:** *"Solution of ordinary differential initial value problems on an Intel hypercube"*. Comput. Math. Applic., **23** (1992), 63-72.
299. **L. Lustman, B. Neta and C. P. Katt:** *"Solution of linear systems of ordinary differential equations on an Intel hypercube"*. SIAM J. Sci. Statist. Comput., **12** (1991), 1480-1485.
300. **J. N. Lyness:** *"Computational techniques based on the Lanczos representation"*. Math. Comp., **28** (1974), 81-123.
301. **A. MacDonald:** *"A semi-Lagrangian and semi-implicit two level integration scheme"*. Mon. Wea. Rev., **114** (1986), 824-830.

302. **A. MacDonald:** *"Accuracy of multiply-upstream, semi-Lagrangian advective scheme"*. Mon. Wea. Rev., **115** (1987), 1446-1449.
303. **N. K. Madsen, G. H. Rodrigue and J. I. Karush:** *"Matrix multiplication by diagonals on a vector parallel processor"*. Information Processing Letters, **5** (1976), 41-45.
304. **G. Majda:** *"Filtering techniques for systems of stiff ordinary differential equations, I"*. SIAM J. Numer. Anal., **21** (1984), 535-566.
305. **G. Majda:** *"Filtering techniques for systems of stiff ordinary differential equations, II. Error estimates"*. SIAM J. Numer. Anal., **22** (1985), 1116-1134.
306. **G. Majda:** *"Stability with large step sizes for multistep discretizations of stiff ordinary differential equations"*. Math. Comput., **47** (1986), 473-502.
307. **G. I. Marchuk:** *"Mathematical models for environmental problems"*. Nauka, Moscow, 1982 (in Russian, but an English translation is also available: North-Holland, Amsterdam, 1986).
308. **G. I. Marchuk and N. N. Yanenko:** *"The application of the splitting-up method (fractional steps) to problems of mathematical physics"*. In: **"SOME PROBLEMS OF NUMERICAL AND APPLIED MATHEMATICS"**, Nauka, Novosibirsk, 1965.
309. **A. Martin:** *"Estimated washout coefficients for sulphur dioxide, nitric oxide, nitrogen dioxide and ozone"*. Atmos. Environ., **18** (1984), 1955-1966.
310. **M. M. Martins:** *"An iterative method of second order for singular systems"*. Report. Department of Mathematics, Univesity of Coimbra, Coimbra, Portugal, 1987.
311. **R. März:** *"Variable stepsize methods"*. In: **"COMPUTATIONAL MATHEMATICS"**, Banach Center Publications, 13, pp. 407-424. Polish Scientific Publishers, Warsaw, 1984.
312. **R. März:** *"On one-leg methods for differential-algebraic equations"*. Circuits Systems Signal Process, **5** (1986), 87-95.
313. **S. A. McKeen, E.-Y Hsie and S. C. Liu:** *"A study of dependence of rural ozone on ozone precursors in the Eastern United States"*. J. Geophys. Res., **96** (1991), 15377-15394.
314. **S. A. McKeen, E.-Y Hsie, M. Trainer, R Tallamraju and S. C. Liu:** *"A regional model study of the ozone budget in the Eastern United States"*. J. Geophys. Res., **96** (1991), 10809-10845.
315. **G. J. McRae, W. R. Goodin and J. H. Seinfeld:** *"Numerical solution of the atmospheric diffusion equation for chemically reacting flows"*. J. Comp. Phys., **45** (1982), 1-42.

316. **J. A. Meijerink and H. A. van der Vorst:** *"An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix"*. Math. Comp., **31** (1977), 148-162.
317. **P. E. Merilees:** *"Numerical experiments with the pseudospectral method in spherical coordinates"*. Atmosphere, **12** (1974), 77-96.
318. **S. G. Michlin:** *"On the rounding error in the conjugate gradients method"*. Vestnik of the Leningrad State University, **19** (1984), 15-23 (in Russian).
319. **W. L. Miranker and I.-L. Chern:** *"Dichotomy and conjugate gradients in the stiff initial value problem"*. Lin. Alg. Appl., **36** (1981), 57-77.
320. **C. R. Molenkamp:** *"Accuracy of finite-difference methods applied to the advection equation"*. J. Appl. Meteor., **7** (1968), 160-167.
321. **C. B. Moler:** *"Iterative refinement in floating point"*. J. Assoc. Comput. Mach., **14** (1967), 316-321.
322. **C. B. Moler:** *"Three research problems in numerical linear algebra"*. In: **"NUMERICAL ANALYSIS"** (G. H. Golub and J. Oliger, eds.), pp. 1-18. American Mathematical Society, Providence, Rhode Island, 1978.
323. **J. Moth and P. C. Hansen:** *"Short-distance communication on the CM-200"*. Internal report. UNI-C (The Danish Computing Centre for Research and Education), Technical University of Denmark, Bldg. 305, DK-2800 Lyngby, Denmark, 1992.
324. **K. W. Morton:** *"Stability and convergence in fluid flow problems"*. Proc. Roy. Soc. (London), **323A** (1971), 227-253.
325. **NAG LIBRARY FOTRAN MANUAL:** *"Mark 7, Vol. 3-4"*. Numerical Algorithms Group, Banbury Road 7, Oxford, England, 1979.
326. **B. Neta:** *"Analysis of finite elements and finite differences for swallow water equations: A review"*. Mathematics and Computers in Simulation, **34** (1992), 141-161.
327. **B. Neta and R. T. Williams:** *"Stability and phase speed for various finite element formulations of the advection equation"*. Computers and Fluids, **14** (1986), 393-410.
328. **S. P. Nørsett:** *"Semiexplicit Runge-Kutta methods"*. Report No. 6-74. Department of Mathematics, University of Trondheim, Trondheim, Norway, 1974.
329. **S. P. Nørsett and P. G. Thomsen:** *"Local error control in SDIRK-methods"*. BIT, **26** (1986), 100-113.
330. **F. Odeh and W. Liniger:** *"Nonlinear fixed-h stability of linear multistep methods"*. J. Math. Anal. Applic., **61** (1977), 308-323.

331. **M. T. Odman and A. G. Russell:** *"Multiscale modeling of pollutant transport and chemistry"*. J. Geophys. Res., **96** (1991), 7363-7370.
332. **M. T. Odman and A. G. Russell:** *"Multiscale horizontal transport for urban and regional air quality modeling"*. In: **"AIR POLLUTION MODELING AND ITS APPLICATIONS IX"** (H. van Dop and G. Kallos, eds), pp. 211-219, Plenum Press, New York, 1992.
333. **P. Oliveira and F. Patricio:** *"Zero-stability of general variable stepsize variable formula methods"*. Manuscript. Department of Mathematics, University of Coimbra, Coimbra, Portugal, 1987.
334. **P. Oliveira and F. Patricio:** *"Linear multistep multiderivative variable stepsize variable formula methods"*. Manuscript. Department of Mathematics, University of Coimbra, Coimbra, Portugal, 1987.
335. **M. P. Olson, J. W. Bottenheim and K. K. Oikawa:** *"Nitrogen source-receptor matrices and model results for Eastern Canada"*. Atmos. Environ., **26A** (1992), 2323-2340.
336. **S. A. Orszag:** *"Transform method for calculation of vector-coupled sums: Application to the spectral form of the vorticity equation"*. J. Atmos. Sci., **27** (1970), 890-895.
337. **S. A. Orszag:** *"Numerical simulation of incompressible flows within simple boundaries, I"*. J. Fluid Mech., **49** (1971), 75-112.
338. **S. A. Orszag:** *"Comparison of pseudospectral and spectral approximations"*. Stud. Appl. Math., **51** (1972), 75-112.
339. **J. Ortega:** *"Introduction to parallel and vector solution of linear systems"*. Plenum Press, New Yourk, 1988.
340. **J. M. Ortega, R. G. Voigt and C. H. Romine:** *"A bibliography on parallel and vector solution numerical algorithms"*. In: **"PARALLEL ALGORITHMS FOR MATRIX COMPUTATIONS"** (K. A. Gallivan, M. T. Heath, E. Ng, J. M. Ortega, B. W. Peyton, R. J. Plemmons, C. H. Romine, A. H. Sameh and R. G. Voigt, eds), pp. 125-197. Society for Industrial and Applied Mathematics, Philadelphia, 1990.
341. **A. Ostermann, P. Kaps and T. D. Bui:** *"The solution of a combustion problem with Rosenbrock methods"*. ACM Trans. Math. Software, **12** (1986), 354-361.
342. **J. Padro:** *"Seasonal contrasts in modelled and observed dry deposition velocities of O_3 , SO_2 and NO_2 over three surfaces"*. Atmos. Environ., **27A** (1993), 807-814.
343. **J. Padro and G. C. Edwards:** *"Sensitivity of ADOM dry deposition velocities to input parameters for SO_2 and NO_2 over three land use types"*. Atmos. Ocean, **29** (1991), 667-685.

344. **J. Padro, G. den Hartog and H. H. Neumann:** *"An investigation of the ADOM dry deposition module using summertime O_3 measurements above a deciduous forest"*. Atmos. Environ., **25A** (1991), 1689-1704.
345. **J. Padro, H. H. Neumann and G. den Hartog:** *"Modelled and observed dry deposition velocity of O_3 above a deciduous forest in the winter"*. Atmos. Environ., **26A** (1992), 775-784.
346. **J. Padro, K. J. Puckett and D. N. Woolridhe:** *"The sensitivity of regionally averaged O_3 and SO_2 concentrations to ADOM dry deposition velocity parameterizations"*. Atmos. Environ., **27A** (1993), 2239-2242.
347. **F. Pasquill and F. B. Smith:** *"Atmospheric diffusion"*. Ellis Horwood Series in Environmental Science, 3rd edition, Ellis Horwood Publishers, Chichester, 1983.
348. **D. W. Pepper and A. J. Baker:** *"A simple one-dimensional finite element algorithm with multidimensional capabilities"*. Numer. Heat Transfer, **3** (1979), 81-95.
349. **D. W. Pepper, C. D. Kern and P. E. Long, Jr.:** *"Modelling the dispersion of atmospheric pollution using cubic splines and chapeau functions"*. Atmos. Environ., **13** (1979), 223-237.
350. **A. G. Petschek and L. D. Libersky.:** *"Stability, accuracy and improvement of Crowley advection schemes"*. Monthly Weather Review, **103** (1975), 1104-1109.
351. **L. R. Petzold:** *"Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations"*. SIAM J. Sci. Statist. Comput., **4** (1983), 136-148.
352. **L. R. Petzold:** *"Order results for implicit Runge-Kutta methods applied to differential-algebraic systems"*. SIAM. J. Numer. Anal., **3** (1986), 837-852.
353. **P. Piotrowski:** *"Stability, consistency and convergence of K-step methods for numerical integration of large systems of ordinary differential equations"*. In: **"CONFERENCE ON NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS"** (J. L. Morris, ed.), pp. 221-227. Springer, Berlin, 1969.
354. **S. Pissanetzky:** *"Sparse matrix technology"*. Academic Press, New York, 1984.
355. **L. P. Prahm and O. Christensen:** *"Long-range transmission of pollutants simulated by a two-dimensional pseudospectral model"*. J. Appl. Meteor., **16** (1977), 896-910.
356. **P. J. Prince and J. R. Dormand:** *"High order embedded Runge-Kutta formulae"*. J. Comput. Appl. Math., **7** (1981), 67-75.
357. **A. Prothero and A. Robinson:** *"On the stability and accuracy of one step methods for solving stiff systems of ordinary differential equations"*. Math. Comput., **28** (1974), 145-162.

358. **J. Pudykiwicz and A. Staniforth:** *"Some properties and comparative performance of the semi-Lagrangian method of Robert in the solution of the advection-diffusion equation"*. Atmos.-Ocean, **22** (1984), 283-308.
359. **D. K. Purnell:** *"Solution of the advection equation by upstream interpolation with a cubic spline"*. Mon. Wea. Rev., **104** (1976), 42-48.
360. **M. Rancic:** *"Semi-Lagrangian piecewise biparabolic scheme for two dimensional horizontal advection of a passive scalar"*. Mon. Wea. Rev., **120** (1992), 1394-1406.
361. **P. J. Rasch and D. L. Williamson:** *"On shape preserving interpolation and semi-Lagrangian transport"*. SIAM J. Statist. Sci. Comput., **4** (1991), 656-687.
362. **J. K. Reid:** *"On the method of conjugate gradients for the solution of large and sparse systems of linear equations"*. In: **"LARGE SPARSE SETS OF LINEAR EQUATIONS"** (J.K.Reid, ed.), pp. 231-254. Academic Press, New York, 1971.
363. **L. F. Richardson:** *"The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam"*. Philos. Trans. Roy. Soc. London, Ser. A, **210** (1910), 307-357.
364. **L. F. Richardson:** *"The deferred approach to limit"*. Philos. Trans. Roy. Soc. London, Ser. A, **226** (1927), 299-361.
365. **R. D. Richtmyer and K. W. Morton:** *"Difference methods for initial value problems"*. Interscience Publishers, New York, 1967.
366. **J. R. Rice and R. F. Boisvert:** *"Solving elliptic problems using ELLPACK"*. Springer, Berlin, 1984.
367. **P. J. Roache:** *"Computational fluid dynamics"*. Hermosa, Albuquerque, New Mexico, 1971.
368. **P. J. Roache:** *"A pseudo-spectral FFT technique for non-periodic problems"*. J. Comp. Phys., **27** (1978), 204-220.
369. **A. J. Robert:** *"A stable numerical integration scheme for the primitive meteorological equations"*. Atmos.-Ocean., **19** (1981), 35-46.
370. **A. J. Robert, T. L. Yee and H. Ritchie:** *"A semi-Lagrangian and semi-implicit numerical integration scheme for multi-level atmospheric models"*. Mon. Wea. Rev., **113** (1985), 388-394.
371. **K. V. Roberts and N. O. Weiss:** *"Convective difference schemes"*. Math. Comp., **20** (1966), 272-290.

372. **H. H. Robertson:** *"Some factors affecting the efficiency of stiff integration routines"*. In: **"NUMERICAL SOFTWARE - NEEDS AND AVAILABILITY"** (D. A. H. Jacobs, ed.), pp. 279-301. Academic Press, London, 1978.
373. **M. Roche:** *"Implicit Runge-Kutta methods for differential algebraic equations"*. SIAM J. Numer. Anal., **26** (1989), 963-975.
374. **R. B. Rood:** *"Numerical advection algorithms and their role in atmospheric transport and chemistry models"*. Rev. Geophys., **25** (1987), 1104-1109.
375. **S. J. Roselle, T. P. Pierce and K. L. Schere:** *"The sensitivity of regional ozone modelling to biogenic hydrocarbons"*. J. Geophys. Res., **96** (1991), 7371-7394.
376. **G. L. Russell and J. A. Lerner:** *"A new finite differencing scheme for the tracer transport equation"*. J. Appl. Meteor., **20** (1981), 1482-1498.
377. **Y. Saad:** *"Variations of Arnoldi's method for computing eigenelements of large unsymmetric matrices"*. Lin. Alg. Appl., **34** (1980), 269-295.
378. **Y. Saad:** *"Krylov subspace methods for solving large unsymmetric linear systems"*. Math. Comp., **37** (1981), 105-126.
379. **Y. Saad and M. H. Schultz:** *"CMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems"*. SIAM J. Sci. Statist. Comput., **7** (1986), 856-869.
380. **A. A. Samarskii:** *"Local one-dimensional difference schemes for multidimensional hyperbolic equations in an arbitrary region"*. Zh. Vychisl. Mat. Mat. Fiz., **4** (1964), 21-25.
381. **J. Sand:** *"On one-leg and linear multistep formulas with variable stepsizes"*. Report TRITA-NA-8112. Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, 1981.
382. **J. Sand:** *"Stability and boundedness of variable-step variable formula methods"*. Report TRITA-NA-8219. Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, 1982.
383. **K. Schaumburg and J. Wasniewski:** *"Use of a semiexplicit Runge-Kutta integration algorithm in a spectroscopic problem"*. Comput. Chem., **2** (1978), 19-24.
384. **K. Schaumburg, J. Wasniewski and Z. Zlatev:** *"Solution of ordinary differential equations with time-dependent coefficients. Development of a semiexplicit Runge-Kutta algorithm and application to a spectroscopic problem"*. Comput. Chem., **3** (1979), 57-63.

385. **K. Schaumburg, J. Wasniewski and Z. Zlatev:** *"The use of sparse matrix techniques in the numerical integration of stiff systems of linear ordinary differential equations"*. Comput. Chem., **4** (1980), 1-12.
386. **K. Schaumburg, J. Wasniewski and Z. Zlatev:** *"Classification of the systems of ordinary differential equations and practical aspects in the numerical integration of large systems"*. Comput. Chem., **4** (1980), 13-18.
387. **K. L. Schere:** *"An evaluation of several numerical advection schemes"*. Atmospheric Environment, **17** (1983), 1897-1907.
388. **R. E. Schlesinger:** *"Effects of upstream-biased third-order space correction terms on multidimensional Crowley advection schemes"*. Monthly Weather Review, **113** (1987), 1109-1130.
389. **W. Schönauer:** *"Scientific computing on vector computers"*. North Holland, Amsterdam, 1987.
390. **P. Seibert and B. Morariu:** *"Improvements of upstream, semi-Lagrangian numerical advection schemes"*. J. Appl. Meteor., **30** (1991), 117-125.
391. **J. H. Seinfeld:** *"Air pollution"*. Wiley, New York, 1986.
392. **A. Semb and E. Amble:** *"Emissions of nitrogen oxides from fossil fuel combusture in Europe"*. NILU - Report 13/81. Norwegian Institute for Air Research (NILU), P. O. Box 130, N-2001 Lillestrøm, Norway, 1981.
393. **L. F. Shampine:** *"Stiffness and non-stiff differential equations solvers"*. In: **"NUMERISCHE BEHANDLUNG VON DIFFERENTIALGLEICHUNGEN"** (J. Li. Morris, ed.). Int. Series Numer. Math., Vol 27, pp. 287-301. Birkhäuser, Basel, Switzerland, 1975.
394. **L. F. Shampine:** *"Stiffness and nonstiff differential equations solvers: II. Detecting stiffness with Runge-Kutta methods"*. ACM Trans. Math. Software, **3** (1977), 44-53.
395. **L. F. Shampine:** *"Evaluation of implicit formulae for the solution of ODE's"*. BIT, **19** (1979), 495-502.
396. **L. F. Shampine:** *"Implementation of Rosenbrock methods"*. ACM Tans. Math. Software, **8** (1982), 93-113.
397. **L. F. Shampine:** *"Efficient extrapolation methods for ODE's"*. IMA J. Numer. Anal., **3** (1983), 383-395.
398. **L. F. Shampine:** *"Stiffness and automatic selection of ODE codes"*. J. Comput. Phys., **54** (1984), 74-86.

399. **L. F. Shampine:** *"The step size used by one-step codes for ODE's"*. Appl. Numer. Math., **1** (1985), 95-106.
400. **L. F. Shampine:** *"Interpolation for Runge-Kutta methods"*. SIAM J. Numer. Anal., **22** (1985), 1014-1027.
401. **L. F. Shampine:** *"Some practical Runge-Kutta formulas"*. Math. Comput., **46** (1986), 135-150.
402. **L. F. Shampine:** *"Diagnosing stiffness for Runge-Kutta methods"*. SIAM J. Statist. Sci. Comput., **12** (1991), 260-272.
403. **L. F. Shampine and L. S. Baca:** *"Fixed versus variable order Runge-Kutta methods"*. ACM Trans. Math. Software, **12** (1986), 1-23.
404. **L. F. Shampine and M. K. Gordon:** *"Computer solution of ordinary differential equations: The initial value problem"*, Freeman, San Francisco, California, 1975.
405. **L. F. Shampine and H. A. Watts:** *"The art of writing a Runge-Kutta code, II"*. Appl. Math. Comput., **5** (1979), 93-121.
406. **L. F. Shampine and H. A. Watts:** *"DEPAC-design of a user oriented package of ODE solvers"*. Report No. SAND-2374. Sandia National Laboratories. Albuquerque, New Mexico, 1980.
407. **L. F. Shampine and H. A. Watts and S. M. Davenport:** *"Solving non-stiff ordinary differential equations: The state of the art"*. SIAM Rev., **18** (1976), 376-411.
408. **L. F. Shampine and W. Zhang:** *"Rate of convergence of multistep codes started by variation of order and stepsize"*. SIAM J. Numer. Anal., **27** (1990), 1506-1518.
409. **A. H. Sherman:** *"On efficient solution of sparse systems of linear and non-linear equations"*. Report No. 46 (Ph D Thesis). Department of Computer Science, Yale University, New Haven, Connecticut, 1975.
410. **A. H. Sherman:** *"Algorithms for sparse Gaussian elimination with partial pivoting"*. ACM Trans. Math. Software, **4**(1978), 330-338.
411. **S. Sillman, J. A. Logan and S. C. Wofsy:** *"Sensitivity of ozone to nitrogen oxides and hydrocarbons in regional ozone episodes"*. J. Geophys. Res., **95** (1990), 1837-1851.
412. **S. Sillman, J. A. Logan and S. C. Wofsy:** *"A regional scale model for ozone in the United States with subgrid representation of urban and power plant plumes"*. J. Geophys. Res., **95** (1990), 5731-5748.
413. **D. Simpson:** *"Long-period modelling of photochemical oxidants in Europe. Model calculations for July 1985"*. Atmos. Environ., **26A** (1992), 1609-1634.

414. **D. Simpson:** *"Photochemical model calculations over Europe for two extended summer periods, 1985 and 1989. Model results and comparisons with observations"*. Atmos. Environ., **27A** (1993), 921-943.
415. **R. D. Skeel:** *"Convergence of multivalued methods for solving ordinary differential equations"*. Ph D Thesis. Department of Computer Science, University of Alberta, Edmonton, Alberta, Canada, 1973.
416. **R. D. Skeel:** *"A theoretical framework for proving accuracy results for deferred corrections"*. SIAM J Numer. Anal., **19** (1982), 170-196.
417. **R. D. Skeel:** *"Construction of variable stepsize multistep methods"*. Math. Comput., **47** (1986), 503-510.
418. **R. D. Skeel:** *"Thirteen ways to estimate global error"*. Numer. Math., **48** (1986), 1-20.
419. **R. D. Skeel:** *"The second order backward differentiation formula is unconditionally zero-stable"*. Report UILU-ENG-86-1753. Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1986.
420. **R. D. Skeel and L. W. Jackson:** *"The stability of variable-stepsize Nordsieck methods"*. SIAM J Numer. Anal., **20** (1983), 840-853.
421. **A. van der Sluis and H. A. van der Vorst:** *"The rate of convergence of conjugate gradients"*. Numer. Math., **48** (1986), 543-560.
422. **A. van der Sluis and H. A. van der Vorst:** *"The convergence behaviour of Ritz values in the presence of close eigenvalues"*. Report No 86-08. Department of Mathematics and Informatics, Delft University of Technology, Julianalaan 132, 2628 BL Delft, Netherlands, 1986.
423. **G. D. Smith:** *"Numerical solution of partial differential equations: Finite differences"*. Oxford University Press, Oxford, 1978.
424. **P. K. Smolarkiewicz:** *"The multidimensional Crowley advection scheme"*. Monthly Weather Review, **110** (1982), 1968-1983.
425. **P. K. Smolarkiewicz:** *"A simple positive definite advection scheme with small implicit diffusion"*. Monthly Weather Review, **111** (1983), 479-486.
426. **P. K. Smolarkiewicz:** *"A fully multidimensional positive definite advection transport algorithm with small implicit diffusion"*. J. Comput. Phys., **54** (1984), 325-375.
427. **P. K. Smolarkiewicz and T. L. Clark:** *"The multidimensional positive definite advection transport algorithm: Further development and applications"*. J. Comput. Phys., **67** (1986), 396-438.

428. **P. K. Smolarkiewicz and W. W. Grabowski:** *"The multidimensional positive definite advection transport algorithm: Nonoscillatory option"*. J. Comput. Phys., **86** (1990), 355-375.
429. **I. D. Sofronov:** *"On the difference solution of the heat flow equation in curvilinear coordinates"*. Zh. Vychisl. Mat. Mat. Fiz., **3** (1962), 786-881.
430. **P. Sonneveld:** *"CGS, a fast Lanzos-type solver for nonsymmetric linear systems"*. SIAM J. Sci. Statist. Comput., **10** (1989), 36-52.
431. **M. N. Spijker:** *"Contractivity in the numerical solution of initial value problems"*. Numer. Math., **42** (1983), 271-290.
432. **M. N. Spijker:** *"A note on monotonicity and contractivity in the numerical solution of stiff differential equations"*. Report 1985Y25. Institute of Applied Mathematics and Computer Science, University of Leiden. Leiden, The Netherlands, 1985.
433. **A. Staniforth and J. Côté:** *"Semi-Lagrangian integration schemes for atmospheric models - A review"*. Monthly Weather Reviews, **119** (1991), 2206-2223.
434. **H. J. Stetter:** *"Analysis of discretization methods for ordinary differential equations"*. Springer, Berlin, 1973.
435. **H. J. Stetter:** *"Local estimation of the global discretization error"*. SIAM J. Numer. Anal., **8** (1971), 512-523.
436. **H. J. Stetter:** *"The defect correction principle and discretization methods"*. Numer. Math., **29** (1978), 425-443.
437. **H. J. Stetter:** *"Global error estimation in Adams PC-codes"*. ACM Trans. Math. Software, **5** (1979), 415-430.
438. **G. W. Stewart:** *"Introduction to matrix computations"*. Academic Press, London-New York, 1973.
439. **G. W. Stewart:** *"The economical storage of plane rotations"*. Numer. Math., **25** (1976), 137-138.
440. **Th. L. van Stijn, J. C. H. van Eijkeren and N. Praagman:** *"A comparison of numerical methods for air-quality models"*. RIVM Report 958702007. National Institute for Public Health and Environmental Hygiene, P.O.Box 1, 3730 BA Bilthoven, The Netherlands, 1987.
441. **W. R. Stockwell, P. Middleton, J. S. Chang and X. Tang:** *"A second generation regional acid deposition model chemical mechanism for regional air quality modeling"*. J. Geophys. Res., **95** (1990), 16343-16367.

442. **J. Stoer:** *"Extrapolation methods for the solution of initial value problems and their practical realization"*. In: **"PROCEEDINGS OF THE CONFERENCE ON NUMERICAL SOLUTION OF ODE'S"** (D.G.Bettis, ed.), pp. 1-21. Springer, Berlin, 1974.
443. **H. S. Stone:** *"High performance computer architecture"*. Addison-Wesley, Reading, 1987.
444. **G. Strang:** *"Linear algebra and its applications"*. Academic Press, New York, 1976.
445. **G. Strang:** *"Wavelet transforms versus Fourier transforms"*. Bulletin of the American Mathematical Society (New Series), **28** (1993), 288-305.
446. **D. Syrakov, G. Djolov and D. Yordanov:** *"Incorporation of planetary boundary Layer dynamics in a numerical model of long-range air pollutant transport"*. Boundary-Layer Meteorology, **26** (1983), 1-13.
447. **G. Söderlind:** *"On the efficient solution of nonlinear equations in numerical methods for stiff ordinary differential equations"*. Report No. TRITA-NA-8114. Department of Computer Science and Numerical Analysis, Royal Institute of Technology, Stockholm, Sweden, 1981.
448. **R. P. Tewarson:** *"Sparse matrices"*. Academic Press, New York, 1973.
449. **R. P. Tewarson:** *"On the solution of sparse non-linear equations and some applications"*. In: **"SPARSITY AND ITS APPLICATIONS"** (D. J. Evans, ed.), pp. 137-152. Cambridge University Press, Cambridge-London, 1985.
450. **P. G. Thomsen and Z. Zlatev:** *"Two-parameters families of predictor-corrector methods for the solution of ordinary differential equations"*. BIT, **19** (1979), 503-517.
451. **W. F. Tinney and W. S. Meyer:** *"Solution of large sparse systems by ordered triangular factorization"*. IEEE Trans. Automatic Control, **18** (1973), 333-346.
452. **C. J. Tremback, J. Powell, W. R. Cotton and R. A. Pielke:** *"The forward-in-time upstream advection scheme: Extension to higher orders"*. Monthly Weather Review, **115** (1987), 540-555.
453. **R. A. Trompert and J. G. Verwer:** *"A static-regridding method for two-dimensional parabolic partial differential equations"*. Appl. Numer. Math., **8** (1991), 65-90.
454. **R. A. Trompert, J. G. Verwer and J. G. Blom:** *"Computing Brine transport in porous media with an adaptive-grid method"*. Report. Centre for Mathematics and Computer Science (CWI), P. O. Box 4079, 1009 AB Amsterdam, The Netherlands, 1992.
455. **R. Vanselow:** *"Nonlinear stability behaviour of linear multistep methods"*. BIT, **23** (1983), 388-396.

456. **R. S. Varga:** *"Matrix iterative analysis"*. Prentice-Hall, Englewood Cliffs, 1962.
457. **M. van Veldhuizen:** *"Consistency and stability for one-step discretizations of stiff differential equations"*. In: **"STIFF DIFFERENTIAL SYSTEMS"** (R. A. Willoughby, ed.), pp. 259-270. Plenum Press, New York, 1974.
458. **M. van Veldhuizen:** *"D-stability"*. SIAM J. Numer. Anal., **18** (1981), 45-64.
459. **A. Venkatram, P. K. Karamchandani and P. K. Misra:** *"Testing a comprehensive acid deposition model"*. Atmos. Environ., **22** (1988), 737-747.
460. **J. H. Verner:** *"Families of embedded Runge-Kutta methods"*. SIAM J. Numer. Anal., **16** (1979), 857-875.
461. **J. G. Verwer, J. G. Blom and J. M. Sanz-Serna:** *"An adaptive moving-grid method for one-dimensional systems of partial differential equations"*. J. Comp. Phys., **82** (1989), 454-486.
462. **J. G. Verwer and M. van Loon:** *"An evaluation of explicit pseudo-steady-state approximation schemes for stiff ODE systems from chemical kinetics"*. Report NM-R9312. Department of Numerical Mathematics, National Research Institute for Mathematics and Computer Science, Kruislaan 413, 1098 SJ Amsterdam, 1993.
463. **J. Vilà-Guerau de Arellano, P. G. Duynkerke, P. J. Jonker and P. J. H. Builtjes:** *"An observational study on the effect of time and space averaging in photochemical models"*. Atmos. Environ., **27A** (1993), 353-362.
464. **P. K. W. Vinsome:** *"Orthomin, an iterative method for solving sparse sets of simultaneous linear equations"*. In: **"PROCEEDINGS OF THE FOURTH SYMPOSIUM ON RESERVOIR SIMULATION"**, pp. 140-159. Society of Petroleum Engineers of AIME, 1976.
465. **V. V. Voevodin:** *"Computational bases of the linear algebra"*. Nauka, Moscow, 1977 (in Russian).
466. **H. van der Vorst:** *"Experience with parallel vector computers for sparse linear systems"*. Supercomputer, **37** (1990), 28-35.
467. **E. L. Washpress:** *"Iterative solution of elliptic systems"*. Prentice Hall, Englewood Cliffs, N. J., 1966.
468. **J. Wasniewski, K. Schaumburg and Z. Zlatev:** *"Vectorizing codes for solving systems of linear ODE's"*. Supercomputer, **18** (1987), 40-51.
469. **J. Wasniewski, Z. Zlatev and K. Schaumburg:** *"A method for reduction of the storage requirements by the use of special computer facilities; application to linear systems of algebraic equations"*. Comput. Chem., **4** (1982), 181-192.

470. **D. S. Watkins and R. W. HansonSmith:** *"The numerical solution of separably stiff systems by precise partitioning"*. ACM Trans. Math. Software, **9** (1983), 293-301.
471. **H. A. Watts:** *"Starting step size for an ODE solver"*. J. Comput. Appl. Math., **9** (1983), 177-191.
472. **H. A. Watts:** *"Stepsize control in ODE solver"*. Trans. Soc. Comput. Sim., **1** (1984), 15-25.
473. **P. A. Wedin:** *"Perturbation bounds in connection with singular value decomposition"*. BIT, **12** (1972), 99-111.
474. **A. Wellburn:** *"Air pollution and acid rain: The biological impact"*. Longman Scientific & Technical, Essex CM20 2JE, England, 1988.
475. **M. Wengle and J. Seinfeld:** *"Pseudospectral solution of atmospheric diffusion problems"*. J. Comp. Phys., **63** (1978), 87-106.
476. **M. L. Wesely:** *"Parameterization of surface resistances to gaseous dry deposition in regional-scale numerical model"*. Atmos. Environ., **23** (1989), 1293-1304.
477. **M. L. Wesely and B. B. Hicks:** *"Some factors that affect the deposition rates of sulphur dioxide and similar gasses on vegetation"*. J. Air Pollut. Control Ass., **27** (1977), 1110-1116.
478. **P. Wesseling:** *"Theoretical and practical aspects of a multigrid method"*. SIAM J. Sci. Statist. Comput., **3** (1982), 387-407.
479. **C. Z. Whitten and M. W. Gery:** *"Development of CBM-X mechanisms for urban and regional AQSM's"*. Report EPA-600/3-86-012, Pb86-155033. U.S. EPA, Research Triangle Park, NC, USA, 1986.
480. **O. Widlund:** *"A Lanczos method for a class of nonsymmetric systems of linear equations"*. SIAM J. Numer. Anal., **15** (1978), 801-812.
481. **H. A. G. Wijshoff:** *"Data organization in parallel computers"*. Kluwer, Dordrecht-Boston, 1989.
482. **J. H. Wilkinson:** *"Error analysis of direct methods of matrix inversion"*. J. Assoc. Comput. Mach., **8** (1961), 281-330.
483. **J. H. Wilkinson:** *"Rounding errors in algebraic processes"*. Notes in Applied Sciences, No. 32, HMSO, London, 1963.
484. **J. H. Wilkinson:** *"The algebraic eigenvalue problem"*. Oxford University Press, Oxford--London, 1965.

485. **J. H. Wilkinson:** *"Modern error analysis"*. SIAM Review, **13** (1971), 548-568.
486. **J. H. Wilkinson:** *"Note on matrices with a very ill-conditioned eigenproblem"*. Numer. Math., **19** (1972), 176-178.
487. **J. H. Wilkinson:** *"Some recent advances in numerical linear algebra"*. In: **"THE STATE OF THE ART IN NUMERICAL ANALYSIS"** (D. A. H. Jacobs, ed.), pp. 3-51. Academic Press, London, 1977.
488. **J. H. Wilkinson:** *"Singular value decomposition-basic aspects"*. In: **"NUMERICAL SOFTWARE-NEEDS AND AVAILABILITY"** D. A. H. Jacobs, ed.), pp. 109-135. Academic Press, London, 1978.
489. **J. H. Wilkinson and C. Reinsch:** *"Handbook for automatic computation: Vol. 2 Linear Algebra"*. Springer, Berlin, 1971.
490. **D. L. Williamson and P. J. Rasch:** *"Two dimensional semi-Lagrangian transport with shape preserving interpolation"*. Monthly Weather Review, **117** (1989), 102-129.
491. **N. N. Yanenko:** *"The method of fractional steps"*. Springer, Berlin, 1971.
492. **G-C. Yang:** *"Paraspice: A parallel direct circuit simulator for shared-memory multiprocessors"*. Ph D Thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1990.
493. **D. M. Young:** *"Iterative methods for solving partial differential equations of elliptic type"*. Trans. Amer. Math. Soc., **75** (1954), 92-111.
494. **D. M. Young:** *"Convergence properties of the symmetric and unsymmetric over-relaxation methods"*. Math. Comp., **24** (1970), 793-807.
495. **D. M. Young:** *"Iterative solution of large linear systems"*. Academic Press, New York, 1971.
496. **D. M. Young:** *"Generalization of Property A and consistent ordering"*. SIAM J. Numer. Anal., **9** (1972), 454-463.
497. **D. M. Young and K. C. Jea:** *"Generalized conjugate gradient acceleration of non-symmetric iterative methods"*. Lin. Alg. Appl., **34** (1980), 159-194.
498. **D. M. Young and D. R. Kincaid:** *"The ITPACK package for large sparse linear systems"*. In: **"ELLIPTIC PROBLEM SOLVERS"** (M. Schultz, ed.), pp. 163-185. Academic Press, New York, 1981.
499. **D. M. Young and D. R. Kincaid:** *"The ITPACK software package"*. In: **"PDE SOFTWARE: MODULES, INTERFACES AND SYSTEMS"** (B. Engquist and T. Smedsaas, eds.), pp. 193-206. North Holland, Amsterdam, 1984.

500. **P. E. Zadunaisky:** "A method for the estimation of errors propagated in the numerical solution of a system of ordinary differential equations". In: "PROC. ASTRON. UNION, SYMPOSIUM NO. 25". Academic Press, New York, 1966.
501. **P. E. Zadunaisky:** "On the estimation of errors propagated in the numerical integration of ordinary differential equations". *Numer. Math.*, **27** (1976), 21-39.
502. **S. T. Zalesak:** "Fully multidimensional flux-corrected transport algorithms for fluids". *J. Comput. Phys.*, **31** (1979), 335-362.
503. **P. Zannetti:** "Air pollution modeling: Theories, computational methods and available software". Van Nostrand Reinhold, New York, 1990.
504. **Z. Zlatev:** "Stability properties of variable stepsize variable formula methods". *Numer. Math.*, **31** (1978), 175-182.
505. **Z. Zlatev:** "On some pivotal strategies in Gaussian elimination by sparse technique". *SIAM J. Numer. Anal.*, **17** (1980), 18-30.
506. **Z. Zlatev:** "On solving some large linear problems by direct methods". Report No. DAIMI PB-111, Department of Computer Science, Aarhus University. Aarhus, Denmark, 1980.
507. **Z. Zlatev:** "Zero-stability properties of the three-ordinate variable stepsize variable formula methods". *Numer. Math.*, **37** (1981), 157-166.
508. **Z. Zlatev:** "Modified diagonally implicit Runge-Kutta methods". *SIAM J. Sci. Statist. Comput.*, **2** (1981), 321-334.
509. **Z. Zlatev:** "Use of iterative refinement in the solution of sparse linear systems". *SIAM J. Numer. Anal.*, **19** (1982), 381-399.
510. **Z. Zlatev:** "Comparison of two pivotal strategies in sparse plane rotations". *Comput. Math. Appl.*, **8** (1982), 119-135.
511. **Z. Zlatev:** "Consistency and convergence of general multistep variable stepsize variable formula methods". *Computing*, **31** (1983), 47-67.
512. **Z. Zlatev:** "Application of predictor-corrector schemes with several correctors in solving air pollution problems". *BIT*, **24** (1984), 700-715.
513. **Z. Zlatev:** "Sparse matrix technique for general matrices: pivotal strategies, decompositions and applications in ODE software". In: "SPARSITY AND ITS APPLICATIONS" (D. J. Evans, ed.), pp. 185-228. Cambridge University Press, Cambridge--London, 1985.

514. **Z. Zlatev:** *"General scheme for solving linear algebraic problems by direct methods"*. Applied Numerical Mathematics, **1** (1985), 176-186.
515. **Z. Zlatev:** *"Mathematical model for studying the sulphur pollution over Europe"*. J. Comput. Appl. Math., **12** (1985), 651-666.
516. **Z. Zlatev:** *"Solving large systems of linear algebraic equations by the use of package Y12M"*. In: **"STRUCTURAL ANALYSIS SYSTEMS"** (A. Niku-Lari, ed.), Vol. 2, pp. 152-160. Pergamon Press, New York, 1985.
517. **Z. Zlatev:** *"Numerical treatment of some mathematical problems describing long-range transport of air pollution"*. In: **"MATHEMATICAL ANALYSIS AND ITS APPLICATIONS"** (S. M. Mazhar, A. Hamoui and N. S. Faour, eds.), pp. 367-383. Pergamon Press, New York, 1986.
518. **Z. Zlatev:** *"Transition to variable stepsize variable formula methods for solving ordinary differential equations"*. In: **"COMPUTATIONAL MATHEMATICS II"** (S. O. Fatunla, ed.), pp. 97-114. Boole Press, Dublin, 1987.
519. **Z. Zlatev:** *"Survey of the advances of exploiting the sparsity in the solution of large problems"*. J. Comput. Appl. Math., **20** (1987), 83-105.
520. **Z. Zlatev:** *"Treatment of some mathematical models describing long-range transport of air pollutants on vector processors"*. Parallel Computing, **6** (1988), 87-99.
521. **Z. Zlatev:** *"Advances in the theory of variable stepsize variable formula methods for ordinary differential equations"*. Applied Mathematics and Computation, **31** (1989), 209-249.
522. **Z. Zlatev:** *"Running large air pollution models on high speed computers"*. Math. Comput. Modelling, **14** (1990), 737-740.
523. **Z. Zlatev:** *"Computations with large and band matrices on vector processors"*. In: **"ADVANCES IN PARALLEL COMPUTING"** (D. J. Evans, ed.), Vol. 1, pp. 7-37. JAI Press, London-Cambridge, England, 1990.
524. **Z. Zlatev:** *"Solving band systems of linear equations by iterative methods on vector processors"*. In: **"NUMERICAL LINEAR ALGEBRA, DIGITAL SIGNAL PROCESSING AND PARALLEL COMPUTING"** (G. Golub and P. Van Dooren, eds.), pp. 713-725. Springer-Verlag, Berlin, Germany, 1990.
525. **Z. Zlatev, E. Berge, J. Christensen, A. Eliassen and Ø. Hov:** *"Long-range transport, critical levels and control strategies"*. In: **"EMEP WORKSHOP ON THE CONTROL OF PHOTOCHEMICAL OXIDANTS IN EUROPE"** (P. Anttila, ed.), pp. 49-54. Finnish Meteorological Institute, Sahaajankatu 22 E, FIN-00810 Helsinki, Finland, 1993.

526. Z. Zlatev and R. Berkowicz: *"Numerical treatment of large-scale air pollution models"*. Comput. Math. Appl., **15** (1988), 93-109.
527. Z. Zlatev, R. Berkowicz, J. Christensen, A. Eliassen, Ø. Hov, T. Iversen and L. P. Prahm: *"An Eulerian air pollution model with non-linear chemistry"*. In: **"PROCEEDING OF THE EMEP WORKSHOP ON THE PROGRESS OF TRANSPORT MODELLING OF NITROGEN COMPOUNDS"** (J. Kluge, ed.), pp. 30-58. EMEP Workshop on the progress of transport modelling of nitrogen compounds, October, 15-18, 1990. Deutscher Wetterdienst, Potsdam, Germany, 1991.
528. Z. Zlatev, R. Berkowicz, J. Christensen, A. Eliassen, Ø. Hov, T. Iversen and L. P. Prahm: *"A quantitative study of the sensitivity of an Eulerian air pollution model with non-linear chemistry to changes of certain physical parameters"*. In: **"PROCEEDINGS OF THE EMEP WORKSHOP ON PHOTOCHEMICAL MODELLING FOR LONG-RANGE TRANSPORT IN RELATION TO ABATEMENT STRATEGIES"** (J. Pankrath, ed.). EMEP Workshop on Photochemical Modelling for Long-Range Transport in Relation to Abatement Strategies (Berlin, April, 16-19, 1991). Umweltbundesamt, Berlin, Germany, 1991.
529. Z. Zlatev, R. Berkowicz and L. P. Prahm: *"Stability restrictions on time-stepsize for numerical integration of first-order partial differential equations"*. J. Comput. Phys., **51** (1983), 1-27.
530. Z. Zlatev, R. Berkowicz and L. P. Prahm: *"Testing subroutines solving advection-diffusion equations in atmospheric environments"*. Comput. Fluids, **11** (1983), 12-38.
531. Z. Zlatev, R. Berkowicz and L. P. Prahm: *"Three dimensional advection-diffusion modelling for regional scale"*. Atmos. Environ., **17** (1983), 491-499.
532. Z. Zlatev, R. Berkowicz and L. P. Prahm: *"Implementation of a variable stepsize variable formula method in the time-integration part of a code for long-range transport of air pollutants"*. J. Comput. Phys., **55** (1984), 279-301.
533. Z. Zlatev, R. Berkowicz and L.P. Prahm: *"Numerical treatment of the advection-diffusion equation. Part IV -Special treatment of the diffusion terms"*. Report No. MST LUFT-A72. Air Pollution Laboratory, Danish Agency of Environmental Protection, Risø National Laboratory, DK-4000 Roskilde, DENMARK, 1983.
534. Z. Zlatev, R. Berkowicz and L.P. Prahm: *"Studying the sulphur pollution over Europe"*. Report No. MST LUFT-A99. Air Pollution Laboratory, Danish Agency of Environmental Protection, Risø National Laboratory, DK-4000 Roskilde, DENMARK, 1985.
535. Z. Zlatev, R. Berkowicz and L.P. Prahm: *"Studying the nitrogen pollution over Europe"*. Report No. MST LUFT-A99. Air Pollution Laboratory, Danish Agency of Environmental Protection, Risø National Laboratory, DK-4000 Roskilde, DENMARK, 1985.

536. **Z. Zlatev and J. Christensen:** *"Studying the sulphur and nitrogen pollution over Europe"*. In: **"AIR POLLUTION AND ITS APPLICATIONS VII"** (H. van Dop, ed.), pp. 351-360. Plenum Press, New York-London, 1989.
537. **Z. Zlatev and J. Christensen:** *"Testing an Eulerian air pollution models with non-linear chemistry"*. Internal report. Ministry of Environment, National Environmental Research Institute, Division for Emissions and Air Pollution, Frederiksborgvej 399, DK-4000 Roskilde, DENMARK, 1990.
538. **Z. Zlatev, J. Christensen and A. Eliassen:** *"Studying high ozone concentrations by using the Danish Eulerian Model"*. Atmos. Environ., **25A** (1993), 845-865.
539. **Z. Zlatev, J. Christensen and Ø. Hov:** *"A Eulerian air pollution model for Europe with nonlinear chemistry"*. Journal of Atmospheric Chemistry, **15** (1992), 1-37.
540. **Z. Zlatev, J. Christensen, J. Moth and J. Wasniewski:** *"Vectorizing codes for studying long-range transport of air pollutants"*. Mathl. Comput. Modelling, **15** (1991), 37-48.
541. **Z. Zlatev, H. B. Nielsen:** *"Solving large and sparse linear least-squares problems by conjugate gradients algorithms"*. Comput. Math. Appl., **15** (1988), 185-202.
542. **Z. Zlatev, K. Schaumburg and J. Wasniewski:** *"Implementation of an iterative refinement option in a code for large and sparse systems"*. Comput. Chem., **4** (1980), 87-99.
543. **Z. Zlatev and P. G. Thomsen:** *"ST: a FORTRAN IV subroutine for the solution of large systems of linear algebraic equations with real coefficients by use of sparse technique"*. Report No. NI-76-05. Institute for Numerical Analysis, Technical University of Denmark, Lyngby, Denmark, 1976.
544. **Z. Zlatev and P. G. Thomsen:** *"Application of backward differentiation methods to the finite element solution of time-dependent problems"*. Internat. J. Numer. Methods Engng., **14** (1979), 1051-1061.
545. **Z. Zlatev and P. G. Thomsen:** *"Automatic solution of differential equations based on the use of linear multistep methods"*. ACM Trans. Math. Software, **5** (1979), 401-414.
546. **Z. Zlatev and P. G. Thomsen:** *"Sparse matrices-efficient decomposition and applications"*. In: **"SPARSE MATRICES AND THEIR USES"** (I. S. Duff, ed.), pp. 367-375. Academic Press, London, 1981.
547. **Z. Zlatev, Ph. Vu, J. Wasniewski and K. Schaumburg:** *"Computations with symmetric, positive definite and band on a parallel vector processor"*. Parallel Computing, **8** (1988), 301-312.

548. **Z. Zlatev and J. Wasniewski:** *"Package Y12M: solution of large and sparse systems of linear algebraic equations"*. Preprint Ser., No. 24. Mathematics Institute, University of Copenhagen. Copenhagen, Denmark, 1978.
549. **Z. Zlatev and J. Wasniewski:** *"Running air pollution models on the Connection Machine"*. Mathl. Comput. Modelling, **20** (1994), 1-17.
550. **Z. Zlatev, J. Wasniewski, J. Moth and M. Venugopal:** *"Optimizing air pollution models on parallel machines"*. Supercomputer, Vol. **9**, No. 2, 1992, 37-48.
551. **Z. Zlatev, J. Wasniewski and K. Schaumburg:** *"A testing scheme for subroutines solving large linear systems"*. Comput. Chem., **5** (1981), 91-100.
552. **Z. Zlatev, J. Wasniewski and K. Schaumburg:** *"Y12M-solution of large and sparse systems of linear algebraic systems"*. Lecture Notes in Computer Science, No. 121. Springer, Berlin, 1981.
553. **Z. Zlatev, J. Wasniewski and K. Schaumburg:** *"Comparison of two algorithms for solving large linear systems"*. SIAM J. Sci. Statist. Comput., **3** (1982), 486-501.
554. **Z. Zlatev, J. Wasniewski and K. Schaumburg:** *"On some useful options in a code for solving stiff systems of linear ordinary differential equations"*. Report 1983-9. Regional Computing Center at the University of Copenhagen, Copenhagen, Denmark, 1983.
555. **Z. Zlatev, J. Wasniewski and K. Schaumburg:** *"Subroutine DENSEI for solving stiff systems of linear ordinary differential equations (basic algorithms, documentation, demonstration programs)"*. Report No. 83-7. Regional Computing Centre at the University of Copenhagen (RECKU), Copenhagen, Denmark, 1983.
556. **Z. Zlatev, J. Wasniewski and K. Schaumburg:** *"Subroutine SPARI for solving stiff systems of linear ordinary differential equations (basic algorithms, documentation, demonstration programs)"*. Report No. 85-6. Regional Computing Centre at the University of Copenhagen (RECKU), Copenhagen, Denmark, 1985.
557. **Z. Zlatev, J. Wasniewski and K. Schaumburg:** *"Exploiting the sparsity in the solution of linear ordinary differential equations"*. Comp. Math. Appl., **11** (1985), 1069-1087.
558. **Z. Zlatev, J. Wasniewski and K. Schaumburg:** *"Numerical treatment of models arising in nuclear magnetic resonance spectroscopy"*. Advances in Engineering Software, **8** (1986), 223-233.
559. **Z. Zlatev, J. Wasniewski and K. Schaumburg:** *"Condition number estimators in a sparse matrix software"*. SIAM J. Sci. Statist. Comput., **7**(1986), 1175-1186.

- 560. Z. Zlatev, J. Wasniewski and K. Schaumburg:** *"Subroutine BAND1 for solving stiff systems of linear ordinary differential equations (basic algorithms, documentation, demonstration programs)"*. Technical Note. Regional Computing Centre at the University of Copenhagen (RECKU), Copenhagen, Denmark, 1988.
- 561. Z. Zlatev, J. Wasniewski and K. Schaumburg:** *"Running conjugate gradient algorithms on three vector machines"*. Supercomputer, **29** (1989), 31-41.
- 562. Z. Zlatev, J. Wasniewski, M. Venogopal and J. Moth:** *"Optimizing air pollution model on two ALLIANT machines"*. In: **"PARALLEL COMPUTATION"**, (A. E. Fincham and B. Ford, eds.), pp. 115-133. Oxford University Press, Oxford-London, 1993.
- 563. O. Østerby and Z. Zlatev:** *"Direct methods for sparse matrices"*. Lecture Notes in Computer Science, No. 157. Springer, Berlin, 1983.

AUTHOR INDEX

A

S. Abarbanel 107, 307
 C. A. Addison 96, 108, 307
 J. Alcamo 17, 18, 307
 R. Alexander 108, 307
 A. P. Altshuler 89, 108, 314
 E. Amble 4, 335
 E. Anderson 83, 84, 212, 223, 307
 K. G. Anlauf 66, 309
 P. Anttila 3, 8, 344
 A. Arakawa 107, 307
 W. E. Arnoldi 84, 307
 W. A. Asman 270, 307, 308
 O. Axelsson 84, 308

B

L. S. Baca 96, 108, 336
 Z. Bai 83, 84, 212, 223, 307
 K. A. Bagrinowski 70, 106, 308
 A. J. Baker 106, 107, 332
 V. B. Balakin 108, 308
 J. A. Balko 66, 308
 C. M. Banic 272
 V. A. Barker 84, 308
 H. M. Barnes 269
 A. Bayliss 75, 107, 308
 K. H. Becker 11, 20, 308
 A. Beguelin 188, 190, 191, 308
 E. Berge 36, 88, 249, 257, 344
 R. Berkowicz 34, 36, 37, 66, 79, 80, 88,
 96, 107, 111, 113, 115, 123, 129,
 130, 133, 136, 140, 145, 150, 151,
 154, 157, 158, 159, 160, 170, 171,
 172, 174, 215, 222, 223, 249, 268,
 269, 309, 322, 324, 345
 R. Bermejo 107, 309
 D. G. Bettis 339
 C. M. Bhumralkar 66, 309
 H. Bildsee 108, 309, 325
 G. Birkhoff 265
 C. H. Bischoff 83, 84, 212, 307
 Å. Björck 77, 96, 108, 309, 313
 J. G. Blom 106, 109, 339, 340
 R. F. Boisvert 84, 215, 333
 D. L. Book 107, 309
 J. P. Boris 107, 309
 P. Borrell 2
 P. M. Borrell 2
 A. Bott 107, 309
 J. Bottenheim 66, 309, 331
 R. K. Brayton 108, 309
 E. O. Brigham 79, 309
 R. Brodzinsky 66, 309
 R. A. Brost 266, 304, 311
 P. K. Brown 95, 310
 K. A. Bruce 66, 309
 J. J. Bufalini 89, 108, 314
 T. D. Bui 97, 326, 331
 P. Builtjes 66, 310, 340
 R. Bulirsh 84, 96, 97, 108, 310

J. R. Bunch 83, 84, 198, 201, 212, 218,
 219, 223, 315
 K. Burrage 96, 142, 173, 310, 311
 Z. Burnstein 96, 107, 310
 J. C. Butcher 92, 93, 96, 142, 145,
 146, 167, 310, 311

C

J. G. Calvert 44, 311
 M. Calvo 173, 311
 B. K. Cantrell 66, 309
 G. R. Carmichael 12, 66, 311
 J. P. Cash 96, 311
 L. A. Chambers 1
 W. L. Chameides 89, 108, 311
 T. F. Chan 95, 311
 J. S. Chang ... 266, 366, 304, 311, 338
 I.-L. Chern 95, 330
 F. N. Chipman 96, 142, 310, 311
 D. P. Chock 74, 107, 311
 J. Christensen 23, 33, 36, 88, 96, 101,
 107, 123, 137, 140, 142, 143, 158,
 172, 190, 222, 223, 230, 249, 257,
 266, 268, 269, 271, 297, 298, 301,
 304, 312, 324, 344, 345, 346
 O. Christensen 66, 80, 107, 312, 332
 A. T. Chronopoulos 108, 312
 T. L. Clark 107, 312, 337
 J. W. Cooley 79, 312
 C. C. Conley 108, 309
 D. S. Conwell 187, 190, 191, 319
 W. R. Cotton 107, 339
 R. Courant 74, 312
 J. Côté 97, 107, 338
 J. J. Coyle 108, 307
 J. Crank 107, 312
 M. Crouzeix 96, 166, 312
 W. P. Crowley 34, 96, 107, 113, 140,
 312
 P. J. Crutzen 44, 314
 C. W. Cryer 96, 108, 312
 A. R. Curtis 96, 108, 312

D

G. Dahlquist 77, 93, 96, 108, 147, 149,
 313, 326
 C. Daly 108, 197, 313
 J. W. Daniel 108, 313
 I. Daubechies 107, 314
 S. M. Davenport 96, 108, 336
 E. S. Davidson ... 186, 190, 191, 327
 T. J. Davis 8
 K. Dekker 108, 167, 314
 K. L. Demerjian 89, 108, 314
 J. W. Demmel ... 83, 84, 212, 223, 307
 R. L. Dennis 266
 F. J. Dentener 44, 314
 R. Derwent 8, 9, 16, 17, 18, 19, 20, 44,
 57, 66, 314

P. Deuflhard 96, 97, 98, 99, 108, 314
 H.-J. Diekhoff 108, 314
 B. Dimitriadis 89, 108, 314
 P. A. M. Dirac 69
 G. Djolov 107, 339
 M. C. Dodge ... 9, 41, 89, 108, 314, 321
 J. J. Dongarra 83, 84, 188, 190, 191,
 197, 198, 199, 201, 202, 204, 205,
 212, 218, 219, 223, 307, 308, 315
 H. van Dop 266, 269, 284, 310, 331, 346
 J. R. Dormant 97, 108, 315, 332
 H. Dovland 3, 4, 315
 P. F. Dubois 84, 216, 315
 J. J. Du Croz 83, 84, 108, 197, 199,
 202, 204, 212, 223, 307, 313, 315
 I. S. Duff 108, 308, 315, 316, 317, 318,
 346
 A. M. Dunker 74, 107, 311
 P. G. Duynkerke 66, 340
 E. G. D'yakonov 106, 107, 317

E

R. C. Easter 107, 317
 A. Ebel 2, 66, 266, 304, 317, 323
 G. C. Edwards 10, 331
 D. Edelson 89, 108, 319
 B. L. Ehle 96, 317
 J. C. H. van Eijkeren 107, 338
 S. C. Eisenstat 84, 108, 190, 197,
 205, 315, 317, 318
 A. Eliassen 8, 14, 23, 33, 36, 41, 47,
 48, 55, 61, 66, 88, 101, 123, 129,
 140, 142, 143, 158, 172, 190, 215,
 230, 249, 257, 267, 268, 269, 271,
 297, 298, 299, 301, 304, 318, 322,
 324, 325, 344, 345, 346
 H. C. Elman 84, 317
 R. M. Endlich 66, 309
 B. Engquist 342
 W. H. Enright 96, 108, 173, 307, 318,
 325
 G. Erbes 107, 324
 A. M. Erisman 108, 316
 D. J. Evans 84, 215, 251, 316, 318,
 319, 339, 343, 344

F

D. K. Fadeev 84, 319
 V. N. Fadeeva 84, 319
 N. S. Faour 344
 L. A. Farrow 89, 108, 319
 S. O. Fatunla 93, 108, 319, 344
 E. Fehlberg 96, 319
 H. Feldmann 66, 304, 323
 B. M. Fellen 108, 173, 325
 A. E. Fincham 348
 B. Ford 348
 B. Forester 118, 123, 129, 319
 B. Fornberg 79, 80, 107, 118, 319
 G. E. Forsythe 108, 223, 319
 D. G. Fox 107, 319
 R. Frank 96, 319

R. Freund 84, 319
 R. F. Freund 187, 190, 191, 319
 K. O. Friedrichs 74, 312

G

P. W. Gaffney 96, 108, 307
 K. A. Gallivan 190, 193, 237, 249, 320,
 331
 F. R. Gantmacher 148, 320
 J. M. Gary 107, 320
 D. Gay 5, 327
 C. W. Gear 92, 95, 96, 97, 108, 142, 146,
 155, 159, 160, 161, 164, 165, 166,
 167, 168, 172, 173, 320
 H. Geiss 66, 304, 323
 A. Geist 188, 190, 191, 308
 J. A. George 108, 321
 M. W. Gary 9, 41, 321, 341
 F. A. Gifford 36, 321
 J. R. Gilbert 108, 249, 322
 Ph. E. Gill 108, 322
 I. Gladwell 96, 108, 307, 322
 R. Glowinski 313
 C. K. Godunov 70, 106, 308
 G. H. Golub 108, 322, 330, 344
 W. R. Goodin 70, 72, 106, 111, 118, 123,
 129, 329
 M. K. Gordon 96, 108, 155, 159, 160,
 167, 168, 173, 336
 D. Gotlieb 75, 80, 107, 307, 308, 322
 N. I. M. Gould 108, 316
 W. W. Grabowski 107, 338
 B. W. Gragg 97, 118, 108, 184, 190,
 322, 328
 A. Grammelvedt 107, 322
 N. Grandjouan 107, 322
 L. D. Grant 298
 P. Grennfelt 16, 17, 18, 19, 20, 44, 66,
 298, 314, 322
 Ph. M. Gresho 107, 322
 A. Greenbaum 83, 84, 212, 216, 223, 307,
 315
 R. D. Grigorieff 166, 322
 R. G. Grimes 84, 108, 316, 326
 R. Guderian 11, 20, 308
 A. Guenter 5, 327
 G. K. Gupta 108, 323
 M. C. Gursky 108, 317, 318
 F. G. Gustavson 108, 190, 197, 205,
 315, 323
 I. Gustavsson 84, 308

H

K. Hain 107, 309
 A. L. Hageman 84, 323
 E. Hairer 92, 93, 96, 97, 98, 108, 142,
 145, 146, 167, 314, 323
 W. D. Hall 107, 312
 N. E. Halvorsen 8, 41, 47, 48, 55, 61,
 66, 267, 325
 S. Hammarling 83, 84, 197, 199, 202,
 204, 212, 307, 315

A. Hamoui 344
 D. A. Hansen 269
 P. C. Hansen 251, 330
 P. M. Hanson 96, 108, 307
 R. J. Hanson 197, 199, 202, 204, 315,
 327
 R. W. HansonSmith 108, 341
 R. M. Harrison 23, 48, 66, 143, 158,
 172, 249, 301, 304, 323
 G. den Hartog 10, 332
 H. Hass 66, 266, 304, 317, 323
 M. T. Heath 108, 190, 249, 320, 321,
 331
 N. Z. Heidam 66, 323
 P. Henrici 90, 97, 323
 O. Hertel 96, 324
 E. Hesstvedt 89, 92, 108, 324
 M. R. Hestenes 84, 215, 324
 B. B. Hicks 10, 324, 341
 N. J. Higham 223, 324
 A. C. Hindmarsh 95, 96, 97, 142, 155,
 159, 160, 168, 310, 324
 E. Holm 107, 324
 L. Hordijk 17, 18, 307
 A. S. Householder 223, 324
 P. J. van der Houwen ... 92, 167, 324
 Ø. Hov 8, 16, 17, 18, 19, 20, 23, 33, 36,
 41, 44, 47, 48, 55, 61, 66, 88, 89,
 92, 96, 101, 108, 123, 129, 140,
 142, 143, 158, 172, 190, 215, 230,
 249, 257, 266, 267, 268, 269, 271,
 297, 298, 301, 304, 314, 318, 322,
 324, 325, 344, 345, 346
 E.-Y. Hsie 66, 329
 T. E. Hull 96, 108, 173, 318, 325
 A. S. Hunding 107, 325
 W. H. Hundsdorfer 97, 325
 W. Hurewicz 108, 325

I

I. S. A. Isaksen 66, 89, 92, 108, 266,
 304, 311, 318, 324
 T. Iversen 8, 36, 41, 47, 48, 55, 61, 66,
 88, 249, 266, 284, 325, 345

J

K. R. Jackson 95, 96, 97, 108, 311,
 318, 325
 L. W. Jackson 97, 108, 337
 D. A. H. Jacobs 334, 342
 H. J. Jacobs ... 66, 266, 304, 317, 323
 J. P. Jacobsen 108, 309, 325
 A. Jaffe 175
 W. Jalby 190, 235, 249, 320
 M. Jankowski 84, 325
 A. J. Jansen 270, 307, 308
 K. C. Jea 84, 342
 H. E. Jeffries 89, 108, 326
 R. Jeltsch 97, 108, 151, 326
 A. Jennings 84, 326
 P. J. Jonker 66, 340

K

G. Kallos 266, 331
 J. Kamari 17, 18, 307
 M. S. Kamel 96, 108, 318
 P. Kaps 97, 326, 331
 P. K. Karamchandani 266, 301, 304, 340
 A. Karp 190, 197, 205, 315
 J. I. Karush 200, 329
 C. P. Katt 184, 190, 328
 P. Kauppi 17, 18, 307
 C. D. Kern 74, 107, 332
 A. A. Khokhar 187, 190, 191, 326
 J. P. Killus 9, 41, 321
 T. Kitada 12
 D. Kincaid 84, 197, 215, 326, 327, 342
 J. Kluge 345
 A. H. Knap 6
 A. N. Kononov 106, 107, 326
 B. Koren 97, 325
 J. F. B. Kraaijevanger ... 108, 326
 H. O. Kreiss 34, 75, 79, 80, 107, 108,
 148, 326
 F. T. Krogh 96, 155, 159, 160, 161, 168,
 172, 173, 197, 326, 327
 D. J. Kuck 186, 190, 191, 327
 Y. Kurihara 108, 327

L

B. Lamb 5, 327
 J. D. Lambert 92, 93, 142, 145, 146,
 167, 327
 P. Lancaster 148, 327
 L. Lapidus 108, 146, 167, 327
 M. Laube 66, 266, 304, 317, 323
 D. H. Lawrie 186, 190, 191, 327
 C. L. Lawson 197, 327
 P. D. Lax 107, 327
 G. K. Leaf 84, 315
 W. R. Leaitch 272
 N. N. Lee 106, 107, 327, 328
 R. L. Lee 107, 322
 S. D. Lee 298
 J. A. Lerner 66, 107, 334
 M. Lescrenier 108, 316
 C. Levit 190, 253, 254, 263, 328
 J. G. Lewis 108, 316
 H. Lewy 74, 312
 L. D. Libersky 107, 332
 B. Lindberg 96, 97, 108, 318, 328
 W. Liniger 96, 97, 108, 313, 330
 J. L. Lions 313
 F. Lisbona 96, 166, 173, 311, 312
 J. W. Liu 108, 321
 S. C. Liu 66, 329
 J. Löbel 11, 20, 308
 J. A. Logan 11, 66, 336
 P. E. Long, Jr. ... 74, 107, 328, 332
 M. van Loon 74, 92, 97, 107, 325, 328,
 340
 P. Lory 108, 314
 Ch. Lubich 97, 108, 323
 B. Lübker 4, 5, 328

C. Ludes 107, 328
 M. Lusia 269
 L. Lustman 184, 190, 328
 J. N. Lyness 75, 328

M

A. MacDonald 66, 107, 272, 328, 329
 S. Madronich 266, 304, 311
 N. K. Madsen 200, 329
 G. Majda 97, 329
 M. A. Malcolm 223, 319
 G. I. Marchuk ... 29, 31, 70, 106, 329
 A. Martin 10, 329
 M. M. Martins 84, 216, 329
 R. März 166, 329
 B. J. Matkowsky 75, 107, 308
 D. R. Matt 10, 324
 S. M. Mazhar 344
 S. A. McKeen 66, 329
 S. A. McKenney 83, 84, 212, 223, 307
 G. J. McRae 70, 72, 106, 111, 118, 123,
 129, 329
 U. Meier 190, 235, 249, 320
 J. A. Meijerink 84, 330
 M. Memmesheimer 66, 266, 304, 317, 323
 P. E. Merilees 107, 330
 W. S. Meyer 108, 339
 S. G. Michlin 84, 330
 P. Middleton 266, 304, 311, 338
 J. H. Miller 309
 M. Minkoff 75, 84, 107, 308, 315
 W. L. Miranker 95, 330
 A. A. Mirin 96, 107, 310
 P. K. Misra 266, 301, 304, 340
 J. L. Moody 6
 C. R. Molenkamp 34, 113, 140, 330
 C. B. Moler 83, 84, 198, 201, 212, 218,
 219, 223, 315, 319, 330
 J. Montijano 173, 311
 R. E. Moore 108, 313
 B. Morariu 107, 335
 J. Ll. Morris 332, 335
 K. W. Morton ... 74, 84, 107, 330, 333
 J. Moth 88, 108, 143, 222, 223, 244,
 249, 251, 257, 330, 346, 347, 348
 W. Murray 108, 322
 S. Mylona 8, 41, 47, 48, 55, 61, 66, 267,
 325

N

E. Ng 108, 190, 249, 320, 321, 331
 B. Neta 75, 184, 190, 328, 330
 H. H. Neumann 10, 332
 O. Nevanlinna 96, 97, 108, 151, 313,
 326
 P. Nicholson 107, 312
 G. von Nieding 11, 20, 308
 H. B. Nielsen 214, 345
 A. Niku-Lari 344
 G. Nordlund 66, 322
 U. Nowak 96, 97, 98, 99, 108, 314

S. P. Nersett 96, 97, 108, 146, 167,
 178, 204, 318, 323, 330

O

H. J. Oberle 108, 314
 F. Odeh 117, 330
 S. Odin 23
 M. T. Odman 106, 107, 109, 115, 140,
 331
 K. K. Oikawa 66, 331
 J. Oliger 34, 75, 79, 80, 107, 148, 326,
 330
 P. Oliveira 173, 331
 M. P. Olson 66, 331
 T. Oppe 84, 326
 S. A. Orszag 74, 75, 79, 80, 107, 319,
 322, 331
 J. M. Ortega 84, 182, 190, 235, 249,
 320, 331
 A. Ostermann 97, 331
 S. Ostrouchov ... 83, 84, 212, 223, 307
 C. J. Ottley 23, 48, 66, 143, 158, 172,
 249, 301, 304, 323

P

P. J. Paes-Leme 166, 322
 J. Pedro 10, 331, 332
 J. Pankrath 284, 317, 345
 F. Pasquill 7, 8, 19, 36, 37, 332
 F. Patricio 173, 331
 C. T. Pedro 108, 312
 T. Peierls 108, 322
 D. W. Pepper 74, 106, 107, 328, 332
 H. C. Perkins 12
 H. Y. J. Pesh 108, 314
 L. K. Peters 12, 66, 308, 311
 A. G. Petschek 107, 332
 L. Petzold 96, 108, 155, 159, 160, 332
 B. W. Peyton 190, 249, 320, 331
 R. A. Pielke 107, 339
 T. P. Pierce 66, 334
 P. Piotrowski 173, 332
 S. Pissanetzky 108, 332
 J. N. Pitts 44, 311
 R. J. Plemmons 108, 190, 193, 249, 320,
 321, 331
 J. Polkinghorne 69
 G. W. Poole, Jr. 108, 316
 S. W. H. Poon 97, 326
 M. Posch 17, 18, 307
 J. Powell 107, 339
 N. Praagman 107, 338
 L. P. Prahm 34, 36, 37, 66, 79, 80, 88,
 107, 111, 113, 115, 123, 129, 130,
 133, 136, 140, 145, 150, 151, 158,
 159, 160, 170, 171, 172, 174, 208,
 212, 215, 222, 223, 249, 268, 269,
 309, 312, 324, 332, 345
 V. K. Prasanna 187, 190, 191, 326
 P. J. Prince 97, 108, 315, 332
 A. Prothero 97, 108, 332

K. J. Puckett 10, 269, 272, 332
 J. Pudykiewicz 107, 333
 D. K. Purnell 107, 333

R

R. Rabe 11, 20, 308
 M. Rancic 107, 333
 P. J. Rash 107, 333, 342
 J. K. Reid 84, 108, 215, 316, 317, 333
 C. Reinsch 223, 342
 P. Rentrop 108, 314
 J. Respass 84, 326
 S. D. Reynolds 266
 L. F. Richardson 84, 97, 333
 R. D. Richtmyer 74, 84, 107, 333
 J. R. Rice 84, 215, 333
 J. Riganati 225
 H. Ritchie 107, 333
 P. J. Roache 75, 107, 333
 A. J. Robert 107, 333
 K. V. Roberts 107, 333
 H. H. Robertson 88, 108, 334
 A. Robinson 97, 108, 332
 M. Roche 108, 334
 G. H. Rodrigue 84, 200, 216, 315, 329
 C. H. Romine 84, 190, 235, 249, 320, 331
 R. B. Rood 107, 334
 D. J. Rose 323
 S. J. Roselle 66, 334
 Ph. M. Roth 266
 E. Runca 17, 18, 307
 A. G. Russell 106, 107, 109, 115, 140, 331
 G. L. Russell 66, 107, 334

S

Y. Saad 84, 95, 97, 108, 219, 320, 334
 R. Sacks-Davis 108, 323
 M. Saeger 89, 108, 326
 J. Saltbones 4, 8, 41, 47, 48, 55, 61, 66, 266, 267, 298, 299, 315, 318, 322, 325
 A. A. Samarskii 84, 334
 A. H. Sameh 186, 190, 191, 193, 225, 249, 303, 315, 320, 327, 331
 J. M. Sanz-Serna 106, 109, 340
 J. Sand 97, 334
 H. Sandnes 8, 41, 47, 48, 55, 61, 66, 267, 325
 M. A. Saunders 108, 322
 K. Schaumburg 97, 108, 210, 212, 216, 222, 223, 309, 325, 334, 335, 340, 346, 347, 348
 K. L. Schere 66, 107, 334, 335
 B. Scherer 284
 J. Schjoldager 66, 298, 322
 R. E. Schlesinger 107, 335
 J. Schneid 96, 108, 319, 326
 T. Schneider 298
 W. Schönauer 182, 190, 335

W. Schöpp 5, 328
 H. Schreiber 11, 20, 308
 E. Schrödinger 69
 M. H. Schultz 84, 108, 219, 317, 318, 334, 342
 U. Schurath 11, 20, 308
 J. A. Scott 108, 317
 A. E. Sedgwick 108, 173, 325
 P. Seibert 107, 335
 P. Seiler 2
 J. H. Seinfeld 5, 19, 37, 70, 72, 106, 107, 108, 111, 118, 123, 129, 146, 167, 327, 329, 335, 341
 A. Semb 4, 335
 R. Seydel 108, 314
 M. E. Shaaban 187, 190, 191, 326
 L. F. Shampine 96, 108, 141, 142, 155, 159, 160, 167, 168, 172, 173, 335, 336
 A. H. Sherman 108, 317, 318, 336
 J. K. Shi 107, 328
 S. Sillman 11, 66, 336
 H. D. Simon 328
 D. Simpson 8, 55, 61, 66, 92, 267, 336, 337
 R. D. Skeel 97, 108, 313, 337
 A. van der Sluis 84, 337
 T. Smedsaas 342
 F. B. Smith 7, 8, 19, 36, 37, 332
 G. D. Smith 74, 84, 337
 P. K. Smolarkiewicz 107, 337, 338
 I. D. Sofronov 107, 338
 P. Sonneveld 84, 338
 D. Sorensen 83, 84, 212, 307
 M. N. Spijker 97, 338
 A. Staniforth 97, 107, 333, 338
 D. H. Stedman 89, 108, 311
 R. S. Stepleman 324
 A. C. Stern 1
 R. Stern 284
 H. J. Stetter 92, 97, 146, 167, 338
 G. W. Stewart 83, 84, 198, 201, 212, 218, 219, 223, 315, 316, 318, 338
 K. Stewart 96, 168, 327
 E. Stiefel 84, 214, 324
 Th. L. van Stijn 107, 338
 W. R. Stockwell 266, 304, 311, 338
 J. Stoer 84, 96, 97, 108, 310, 339
 H. S. Stone 191, 339
 F. Stordal 66, 318
 G. Strang 107, 223, 339
 V. Sunderam 188, 190, 191, 308
 R. N. Swarztrauber 221
 D. Syrakov 107, 339
 G. Söderlind 96, 97, 108, 313, 339

T

R. Tallamraju 66, 329
 X. Tang 266, 338
 C. Temperton 221
 Th. W. Teshe 266
 P. E. Tescher 108, 323
 R. P. Tewarson 108, 339
 S. Tilly 4, 328

D. T. Tingey 11,20,308
 W. F. Tinney 108,339
 P. G. Thomsen 96,108,168,318,330,
 339,346
 M. Trainer 66,329
 C. J. Tremback 107,339
 R. A. Trompert 106,109,339
 K. W. Tu 97,108,166,320
 J. W. Tukey 79,312
 E. Turkel 107,307

U

C. W. Ueberhuber 96,319
 W. Ulrich 107,328

V

P. Van Dooren 344
 C. F. Van Loan 108,322
 R. Vanselow 173,339
 R. S. Varga 84,215,340
 M. van Veldhuizen 108,340
 A. Venkatram 266,301,304,340
 M. Venugopal 108,244,249,347,348
 J. H. Verner 108,340
 J. G. Verwer 92,97,106,108,109,
 167,314,325,339,340
 J. Vilà-Guerau de Arellano 66,340
 R. K. W. Vinsome 215,216,340
 V. V. Voevodin 223,340
 R. G. Voigt ... 84,190,249,320,331
 H. A. van der Vorst 84,330,337,340
 Ph. Vu ... 108,210,212,216,222,346

W

H. M. Wagner 11,20,308
 C. J. Walcek 266,304,311
 J. Walsh 88
 C.-L. Wang 187,190,191,326
 G. Wanner 92,93,97,98,108,142,
 145,146,167,323
 E. L. Washpress 107,215,340
 J. Wasniewski 88,97,108,140,143,
 197,210,212,216,222,223,244,
 249,257,264,315,334,335,340,
 346,347,348
 D. S. Watanabe 97,108,161,164,165,
 166,173,320
 D. S. Watkins 108,341
 G. A. Watson 313
 H. A. Watts ... 96,108,160,336,341
 P. A. Wedin 223,341
 N. O. Weiss 107,333
 A. Wellburn 2,5,10,11,20,341
 B. Wendroff 107,327
 M. Wengle 107,341
 Ml.L. Wesely 10,341
 P. Wesseling 84,341
 H. Westberg 5,327
 D. M. Whelpdale 6

J. R. Whiteman 318
 C. Z. Whitten 9,41,321,341
 O. Widlund 84,341
 H. A. G. Wijshoff 183,341
 J. H. Wilkinson 223,341,342
 R. T. Williams 75,330
 D. L. Williamson 107,333,342
 R. A. Willoughby 313,323,340
 S. C. Wofsy 11,66,336
 C. J. Wolters 298
 D. N. Woolridhe 10,332
 H. Wozniakowski 84,325
 M. H. Wright 108,322
 M. Wulkow 96,97,98,99,108,314

Y

N. N. Yanenko 106,329,342
 G-C. Yang 108,342
 T. L. Yee 107,333
 D. M. Young 84,215,323,326,384,342
 D. Yordanov 107,339

Z

P. E. Zadunaisky 97,343
 S. T. Zalesak 107,343
 P. Zanneti 19,21,343
 W. Zhang 96,108,141,336
 Z. Zlatev 13,23,33,34,36,37,48,66,
 79,80,84,88,96,101,107,108,
 111,113,115,123,129,130,133,
 136,140,142,143,145,150,151,
 153,154,158,159,160,161,162,
 163,154,165,168,170,171,172,
 173,174,190,210,212,214,215,
 216,220,221,222,223,230,244,
 249,257,263,264,266,267,268,
 269,271,297,298,301,304,309,
 323,324,334,335,339,340,342,
 343-348
 M. Zugch 96,97,108,314

Ø

O. Østerby 108,348

SUBJECT INDEX

A

Absolute stability
 control 157, 159, 166
 intervals 149-155
 regions 147, 148
 requirements 160, 169
Acceptable levels 1, 14, 15, 20, 305
 approximate 15
 definition of 15
 exact 15
Accuracy
 control 159, 166
 order of 151, 155, 167
 requirements ... 159, 160, 169, 207
Advection 1, 2, 7, 21, 28, 112, 227, 239
 horizontal 23, 72, 87, 106, 107, 227
 sub-model 143, 156, 194, 227, 238, 259
 vertical 104
Air pollutants
 advection of 1, 2
 concentration of 1
 deposition of 1, 2, 10
 diffusion of 2, 8
 emission of 1, 2, 3
 primary 2
 secondary 2
 long-range transport of 2, 21, 22
 need for reduction 12
 reduction of ... 1, 12, 13, 22, 296

B

Bandedness 82, 193, 201, 212
BLAS 197, 199, 200, 202, 204, 218
Boundary conditions 72, 82, 100, 101
 Dirichlet ... 82, 102, 103, 132, 136
 horizontal 102
 periodic 75, 76, 102, 118
 vertical 102
 von Neumann 102
Boundary mixing layer 7

C

Cache memory 182, 183, 186, 189, 226, 237, 243
CEDAR 186, 187, 189, 191
Chaining 181
Characteristics 74, 88, 107
Chemical reactions 2, 3, 8, 41, 121, 141, 222, 227, 239, 248, 258
 ammonia-ammonium 41, 44, 47
 equilibrium of 47, 48
 linear 9, 44, 170
 mathematical description of 49
 non-linear 44, 244, 268
 optimal set of 8, 9, 41
 photochemical 44, 45, 46, 51, 244,
 rate of 45, 46

Chemical scheme 9, 41
 condensed CBM IV 9, 41, 44, 47, 51, 96, 280
 optimal 9, 44
 simplified 44, 45, 49, 50, 51, 244, 258
Chemical sub-model 142, 145, 166, 195, 223, 227, 239, 259
Chemistry 21
 ammonia-ammonium 47
Clusters of processors 186, 187, 191
Computers
 heterogeneous 187, 188, 189, 191
 high-speed 175-190
 massively parallel 100, 185-188, 251
 MIMD 177
 parallel 100, 107, 175-190, 209, 221, 244, 248
 PVM 188
 scalar 175
 sequential 107, 175
 SIMD 177
 UNIX 188
 vector 107, 175-190, 193-223, 238
Concentration
 acceptable levels of ... 1, 239
 background 103
 critical levels of 1
Cone rotation 113
Conservation law 29
Control strategies ... 14, 15, 16, 19
 second generation 19
Cosine hill rotation 113
Critical levels ... 1, 11, 14, 15, 20
 approximate 15
 definition of 15
 exact 15
CSCFM 161, 163, 164, 171, 173

D

Data locality 182
Deposition 1, 2, 10, 21, 22, 39, 70, 100, 108
 acceptable levels of 299
 dry 2, 10, 39, 72, 100, 102
 in Denmark 13, 14
 sub-model 100, 194, 195, 232, 240
 sulphur 13
 total 14
 wet 2, 10, 39, 100
Derivatives
 spatial 39, 54, 55
 time 55
Diffusion 2, 7, 8, 21, 35, 53, 58, 70, 86, 87, 88, 104
 artificial 8
 horizontal 18, 37, 38, 86, 87, 107
 mathematical description of 8
 sub-model ... 86, 194, 195, 280, 291
 vertical 37, 104

Direct methods ... 211,212,219,223
 Cholesky 83,84
 Gaussian elimination ... 93,195

E

Economical cost 12,14
Effects 2,11,12,15,16,17,18,19,20
 direct 11
 indirect 11
 on animals 11
 on humans 11
 on plants 11
Emission 1,2,3,21,51
 ammonia 329
 data 3
 increase of 27
 inventories for 3,4
 reduction of 26
 nitrogen (NO_x) 16,329
 sulphur 13,329
 VOC 329
Emission sources 1,2,3
 anthropogenic 2,3,4
 human-made .. see anthropogenic
 man-made see anthropogenic
 natural 2,3
Episode 325,360
Equations
 linear algebraic 193,197,211,
 216,219-223
 linear ordinary differential 40,
 95,109,121
 partial differential 21,22,28
 ordinary differential 22,39,49
Error
 common 61
 compensating 64,65
 discretization 61,62,145
 estimated 98,106,173
 estimation of the 89,99,108,159,
 160,167,263
 global 97,166,167,173
 Lagrangian 61
 local 97,167,173
 tolerance 159,160,166
 truncation 97,167,173
Error estimator
 automatic 96
Extrapolation 88,89,97,98,99,108

F

Factor(s) of a matrix
 Cholesky 83,84
 triangular 83,84,220
Factorization
 Cholesky 83,84
 Gaussian elimination ... 93,212
 Forester 118,123
 passive 118,123
Fourier transforms 75,220
 fast 79,193-196,220,222

multiple 221,223
 vectorized 220,223
Free troposphere 103

G

GFLOPS 19
Grid
 equidistant 23,24,74,76,107,144,
 146
 local refinement of 75,105,106
 non-equidistant see variable
 spatial 23,76,107,143
 temporal 146,162
 variable 104,162
Gaussian elimination 93,212

H

Harwell subroutines 154
Heterogenous networks 187,188,189,
 191
Horizontal planes 71,72,104
Hydroxyl radical 359

I

Initial value condition 32,52,76,
 101,108
Integration methods
 Adams 164
 Backward Euler 93,99
 CSCFM 161,163,164,171,173
 explicit 145,146,153
 extrapolation 88,89,97,108
 for stiff systems 89,93
 Forward Euler 90,91
 implicit 93,94,98,145,166
 linear multistep 146-153,163,165
 leap frog rule 152,153
 mid-point rule
 explicit 152
 implicit 98
 optimal set of 170,171
 predictor-corrector 148,152-173
 Runge-Kutta 89
 starting procedure of 146,152
 trapezoidal rule 98
 two-ordinate formulae 164-166
 VSVFM 161-172
Integration step 167
Interconnections 16,17,18
Inventories 3,4,5
Iterative methods,
 BICGSTAB 95
 Block-Gauss-Seidel 96
 CG (conjugate gradients) 211,212
 CGS 95
 conjugate-gradient-type ... 219
 Gauss-Seidel 96
 GMRES 95,219

Jacobi method 96, 215
orthomin 216
preconditioned 95, 211, 215, 216,
 218
simple iteration 93
TFQMR 95

J

Jacobian matrix 94, 95, 96, 166
shifted 94, 95

K

Kernels 190, 197, 205, 209, 212

L

LARKIN 97, 99
LAPACK 212
Leap-frog rule 152, 153
Levels
 acceptable 1, 6, 12, 14, 15, 20
 critical ... 1, 6, 12, 14, 15, 16, 20
Linear algebra operations,
 basic 195-197, 220, 222
LINPACK 198, 201, 202, 212-219
Local refinement 75, 105, 106, 109
 dynamic 105
 static 105
Long-range transport 1, 2, 6, 7, 9, 14,
 17
Loss terms 51, 90, 278

M

Matrix
 anti-symmetric 148
 banded 82, 83, 193-223
 block-diagonal 80, 144
 dense 82, 197, 198, 199, 222
 diagonal ... 80, 144, 146, 148, 149
 diagonally dominant 96
 eigenvalues of ... 147, 148, 149
 general 95
 identity 82, 87, 104
 inverse of 82, 83, 84
 lower triangular 220
 orthogonal 83
 permutation 80, 144
 positive definite 82, 83, 87, 104,
 115, 195, 197, 201, 211, 218
 sparse 198
 symmetric 82, 83, 87, 104, 115, 195,
 197, 201, 211, 218
 triangular 83, 220
 test 200, 201, 204, 212, 222
 tri-diagonal 85, 313
 upper triangular 83, 220
Measurement stations 34

Measurements 33-35
Methods

explicit ... 145, 146, 153, 194, 220
 finite differences 74, 81, 82, 87,
 104, 194, 220
 finite elements 74, 75, 81, 82, 85,
 87, 104, 118, 195, 219, 222, 285,
 313
 Fourier 75, 80, 81
 implicit 93, 94, 98, 145, 166, 194,
 219
 of characteristics ... 74, 88, 107
 pseudospectral 75, 76, 80, 86, 87,
 102, 104, 115, 144, 146, 194, 220
 semi-Lagrangian 74, 107
 spectral expansions 74, 75
 spectral 75
MFLOPS 176, 196-198, 204
Mixing height 7, 131
Models
 air pollution ... 1, 3, 7, 8, 14, 20
 choice of 41, 60
 Eulerian 24, 54, 55, 66
 general description of 53
 grid 55
 Lagrangian 54-65
 mathematical ... 1, 6, 14, 16, 19, 53
 one-dimensional 31, 32
 reliable 25, 27, 65
 three-dimensional 25, 29, 53, 284,
 292
 trajectory 55
 two-dimensional ... 25, 30, 54, 194
 use of 26
Multiprocessors 209, 210
Multitasking 209

N

NAG Library 212, 221
NAG Subroutines 221
Nitrogen di-oxide 332, 334, 336-339,
 345, 347, 353, 357, 363, 365
Nitrogen oxides 4, 5
Numerical methods see methods
Newton iteration 94
 classical 94, 95
 quasi 94
 modified 94

O

Operators
 Laplace 201, 212
Ozone 358
 high concentrations of 360
 reduction of 359

P

Periodicity 75

Pipelining 179,232,233,235
Plume rotation 115
Pollutants
 primary 2,3,8,9
 secondary 2,3,8,9
Precipitation 10
Precipitation scavenging 10
Preconditioner 95,216,219
Production terms 51,53,90,278
Pseudospectral discretization 75,
 80,86,87
Puff rotation 115
PVM 188

R

RAINS 17,18
Rate coefficients 45,46
Reactions
 chemical 8,22,25,28,41,44,45,49
 linear 9
 non-linear 44
 photochemical 44,45,46,51
Receptor 51,53,62
Reduction of air pollution 1,6,12
 optimal 1
Rotation test
 cone 113
 cosine hill 113
 puff 115
 plume 115

Q

Quasi-steady-state-algorithm 89,
 90,92,108

S

Smoothing 298,314
Solution
 analytical 33,34,54
 exact 33
 numerical 32,33
Space domain 23,24,30
Sparsity 198,205,207,210,212,217,
 222
Speed of computations 196,197,203,
 207,212,215,221
Splitting 70,71,72,106
 according to physical processes
 70,71,106
 advantages of 72,106
 to one dimensional sub-models
 106
Stability 145,155,159,172
 bounds 148
 control 157,166,172
 requirements ... 145,154,159,169
Stiff systems ... 88,92,93,142,145
Stopping criteria 102

Storage scheme 246
 for band matrices
 by diagonals 202,203,218
 LINPACK 202
 sparse 202,203,218
 for dense matrix 202
Sulphur deposition 13,14
Sulphate 340-342,350,352
Sulphur di-oxide 332,333,337-339,
 345,346,350
Sulphur emission sources ... 13,14
Sulphur pollution 8,9,16,17
Symmetry 212
System of PDE's 22,28,35,39,54,55,
 63,72,193,198
Systems of ODE's 41,49,51,55,56,
 79,83,88,92,97,194,219
 linear 79,93,100,108,194
 stiff ... 88,92,112,142,145,195

T

Test-problems 33,34
TFLOPS 19
Trajectory 55,56,57,60,61,63
 backward 55,57
 forward 55,57
 wind 55,112,113
Transport 1,2,6,22,28
Trigonometric polynomials 75,79,87
Tropopause 7
Truncation error 167,173

U

Unrolling 181,182,204-218,222
UNIX 188
Units 46,51

V

Variable time-stepsize 154,161,172
Vector processors 177-190
Vector registers 178-181
Vertical lines 71
VSVFM 161-172

W

Wind 1,7,22,23,29,32,35,59,105,112
Wind fields 160,343
Wind velocities 23,28,29,32,33,73,
 82,113,144,145,153,156,157,
 166,171

Z

Zero-stability 161,163,164,165,173

Environmental Science and Technology Library

1. A. Caetano, M.N. De Pinho, E. Drioli and H. Muntau (eds.), *Membrane Technology: Applications to Industrial Wastewater Treatment*. 1995
ISBN 0-7923-3209-1
2. Z. Zlatev: *Computer Treatment of Large Air Pollution Models*. 1995
ISBN 0-7923-3328-4