Numerical Methods in Smog Prediction

Numerical Methods
in
Smog Prediction

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam,
op gezag van de Rector Magnificus,
prof. dr. P.W.M. de Meijer
ten overstaan van een door
het college van dekanen ingestelde commissie
in het openbaar te verdedigen in de Aula der Universiteit
op maandag 17 juni 1996 te 15.00 uur

door

Martin van Loon

geboren te Amsterdam

CWI
Amsterdam
1996

Numerieke methoden in smogvoorspelling

Promotor: Prof. Dr. P.J. van der Houwen
Co-promotores: Dr. J.G. Verwer en Dr. F.A.A.M. de Leeuw
Faculteit: Wiskunde, Informatica, Natuurkunde en Sterrekunde

Het hier beschreven onderzoek is mede mogelijk gemaakt door financiële steun van het Rijksinstituut voor Volksgezondheid en Milieuhygiène voor het project EUSMOG.

Voor mijn ouders

Voorwoord

Dit proefschrift is het resultaat van vier jaar onderzoek, gedaan op het Centrum voor Wiskunde en Informatica (CWI). Het is een leerzame, maar zeker ook plezierige periode geweest, waar ik met genoegen op kan terugzien. Daarom wil ik van deze gelegenheid gebruik maken alle collega's van de afdeling Numerieke Wiskunde te bedanken voor hun bijdrage aan de goede, ontspannen werksfeer op "onze" afdeling.

Graag spreek ik op deze plaats mijn bijzondere dank uit aan het adres van mijn begeleider, Dr. J.G. Verwer, voor de stimulerende wijze waarop hij het project EUSMOG heeft geleid.

Dr. F.A.A.M. de Leeuw van het RIVM dank ik voor de begeleiding en samenwerking met betrekking tot de atmosferiche en chemische aspecten van het project.

Prof. P.J. van der Houwen dank ik voor het feit dat hij het uit dit project voortgekomen proefschrift wilde "promoten".

Iemand die zeker hier genoemd moet worden, al vindt ze dat zelf wellicht onnodig, is Joke Blom. Ik kijk met plezier terug op de vele gesprekken en discussies over allerlei onderwerpen gedurende de periode waarin we kamergenoten waren. Haar nooit aflatende bereidwillige opstelling als on-line manual op met name programmeergebied is van grote waarde geweest voor de voortgang van het project.

Als ik terug kijk op de afgelopen vier jaar, dan mag ik veel zegeningen tellen. Niet alleen vanwege de gezondheid en kracht om dit project te beginnen en af te ronden, maar ook vanwege voorspoed en geluk in mijn privé-situatie. Ik dank Tineke en Mariska voor de vreugde die zij mij hebben gegeven. Bovenal gaat mijn dank uit naar God die alles maakte en ook met mij alles wèl wilde maken.

Maarten van Loon

Contents

Chapter 1

Introduction

1.1 Smog

The word *smog* is a combination of *smoke* and *fog* and was originally used to describe city fogs containing large amounts of air toxics [14]. The classic example of this kind of air pollution is the notorious London smog episode of December 1952, causing 4000 excess deaths. In contrast to the 1950s and 1960s, smog problems nowadays are not restricted to large cities with heavy traffic (like Los Angeles) but has become a wide spread phenomenon. The word smog now stands for increased concentrations of certain health damaging pollutants. In The Netherlands, a distinction is made between winter smog episodes, characterized by high levels of sulphur dioxide concentrations (SO_2) and dust, and summer smog episodes, characterized by high levels of ozone concentrations (O_3). Despite world-wide concern about ozone gaps in the atmosphere, high ozone concentrations at ground level are harmful not only for human beings but also for animals and vegetation. With respect to human health, exposure to high ozone concentrations may cause breath problems and even lung diseases. The same holds for exposure to SO_2. Unlike SO_2, ozone is not emitted by any source, neither anthropogenic nor natural. As we will see in Chapter 2, Section 2.3, where the chemical model is described, ozone is formed by reaction chains starting with a reaction of an organic compound with the OH radical. Therefore, these organic compounds are called *precursors*. They are emitted by anthropogenic as well as natural sources, like isoprene that is emitted by forests.

1.2 Predicting smog

One of the tasks of the Dutch National Institute of Public Health and Environmental Protection (RIVM) is to provide the local authorities with expected forecasts for levels of air pollution. In case smog is expected, measurements can be taken and the public can be informed. To produce such a forecast, a smog prediction model is necessary. Presently (1996) an Eulerian grid model, EUROS, is used for winter smog episodes. The model runs on a workstation. In the winter, it is started every morning automatically and is supposed to deliver its output within a few hours. Because in EUROS only 5 species are taken into account and the chemical interaction between the species is slow,

an Eulerian grid model can perform the task of predicting SO_2 concentrations within an acceptable amount of CPU time on a workstation. For summer smog a Lagrangian type of model is used which is also used on a routine daily basis. Until now, different types of models are applied because summer smog forecasting requires a detailed atmospheric chemistry leading to ozone formation, involving many species. Not considering chemistry, the total amount of CPU time for a model run is linear in the number of species. This makes an Eulerian summer smog model already a number of times more expensive than a winter smog model. Apart from that, the system of ordinary differential equations (ODEs) arising from summer smog chemistry is stiff, in contrast to winter smog chemistry, which only involves a (slow) transformation of SO_2 into SO_4. Because of the stiffness of the summer chemistry, its numerical evaluation is a time-consuming process.

The purpose of the project EUSMOG is the development of an Eulerian grid model for summer smog prediction, that is also capable of producing its forecasts within a few hours. The major result of this project is the research and the summer smog model implementation reported in this thesis.

1.3 The project EUSMOG

In 1993 the department of numerical mathematics at CWI and the Air Laboratory of the National Institute of Public Health and Environmental Protection (RIVM) started their cooperation within the project EUSMOG. For CWI the purpose of this project was the development of new numerical techniques for implementation in the successor of EUROS, which got the temporary name CWIROS. During the course of the project, however, the name CWIROS has not been changed. CWIROS should be able to perform the same tasks as EUROS, but now for both winter and summer smog prediction using one and the same Eulerian grid model. The reason to aim at this goal, which seemed unachievable before, is that nowadays computers have become much faster and an Eulerian grid model for summer smog has come in reach of a workstation, provided that the numerics are handled as efficiently as possible and the chemical reaction mechanism does not require the modeling of too many species. From the description of the chemical mechanism in Section 2.3 it can be concluded that a mechanism has been formulated that meets the latter requirement. Only 17 reactions between 15 species are involved and the chemical mechanism is only moderately stiff. Reducing chains of reactions into one reaction is the reason for the limited number of species. Because of this reduction, intermediate radicals are not present, which explains the moderate stiffness. Yet, the essential characteristics of ozone formation seem to be retained if we look at the model results presented in Chapter 7. Of course, modeling the complex atmospheric chemistry in such a compact way introduces modeling errors, which are probably sometimes large in parts of the model domain. However, a compromise must be made between detailed modeling and restricting the operational computation time of the resulting model. Not only for the chemistry a compromise had to be made, but for

other physical processes as well. For example, the vertical stratification is modeled by only four layers and the vertical diffusion has been parametrized (see Chapter 2).

All physical and chemical modeling aspects have been the responsibility of the RIVM. The model description (see Chapter 2) is the starting point for the research carried out at CWI. During the course of the project no essential changes in the modeling have been carried through. Only minor adjustments in parametrizations etc. have been done, sometimes on CWI's proposal.

1.4 This work

The task of CWI in the project EUSMOG consisted of the development and implementation of new and existing numerical methods for CWIROS. Because application of grid refinement (see below) was one of the numerical techniques to be implemented in CWIROS, the decision was made not to update EUROS, because the datastructure, required for the grid refinement, would not easily fit in the existing code. Hence, CWIROS has been built up from scratch using building blocks from EUROS. Implementing CWIROS was also the task of CWI within the project. Both CWI tasks have been carried out by the author of this thesis. The model implementation was the author's own responsibility, the research reported here is partly based on joint work with various colleagues at CWI. Key issues of the research were

- Grid refinement: a local uniform grid refinement method has been developed, based on earlier work at CWI [63, 7, 6], for application to the model. The technique has been adapted for finite volume grids in spherical coordinates. The original technique uses the grid point approach. The latter is a disadvantage because it does not preserve mass in a natural way. Since mass conservation is important for the present application, the finite volume approach has been followed. Also the original datastructure has been adapted.
  The grid refinement technique offers the possibility to refine the grid dynamically in areas with large solution gradients. A priori chosen automatic refinement in areas of user interest is also possible. The results of the model runs clearly show that higher resolution is needed for summer smog prediction. The refinement technique offers higher resolution where necessary. Where no refinement is needed, only computations on the coarse base grid are done. In this way a considerable amount of CPU time is saved compared to the situation in which the whole model domain is covered with a fine grid.

- Advection schemes: numerical algorithms for advection have been studied. The second-moment method which was implemented in EUROS, has some disadvantages that made it impossible to use this scheme in CWIROS. One is the large storage requirement. Another drawback is that it is unclear how to combine this advection scheme with the other processes in the model, in particular chemistry. We also note that

the second-moment method does not prevent under- and overshoots although the numerical solutions are guaranteed to be positive. Finally, the scheme is not easily applicable to the sequence of the grid structures produced by the refinement technique.

- Solution methods for chemical kinetics problems: numerical treatment of chemical kinetics is the computationally most intensive part of the model. Therefore, considerable research effort has been put into fast and efficient methods for solving ODEs arising from chemical kinetics. State-of-the-art methods as well as special purpose solvers are considered for application in the model. In particular, we have tried to find ways to accelerate existing state-of-the-art and special purpose solvers.

1.5 Mathematical framework

In atmospheric models, it is usual to apply *operator splitting* or the method of *fractional steps*. This approach was also followed in EUROS. A detailed description including analysis can be found in [46]. Since we apply it also in CWIROS, we describe it here shortly.

The full atmospheric model equation in spherical coordinates can be written as

$$\frac{\partial c(\phi,\theta,z,t)}{\partial t} = F_1(c,\phi,\theta,z,t) + \ldots + F_N(c,\phi,\theta,z,t), \tag{1.1}$$

where c denotes the concentration vector, ϕ, θ and z the spatial coordinates and t the time. The functions F_i, $i = 1,\ldots N$, represent the physical and chemical processes that are modeled. The full equation of the model described in this thesis (see Section 2.2.2) clearly is of the form (1.1). Instead of integrating equation (1.1) at once, in the operator splitting approach the integration is done for each process separately. In the present way of application, this means that the following sequence of differential equations is solved over the time interval $[t_0, t_1]$

$$\begin{cases} \dfrac{\partial c_i(\phi,\theta,z,t)}{\partial t} = F_i(c_i,\phi,\theta,z,t), & \text{for } i = 1,\ldots,N, \\ c_i(\phi,\theta,z,t_0) = c_{i-1}(\phi,\theta,z,t_1), \end{cases} \tag{1.2}$$

with $c_0(\phi,\theta,z,t_1) = c(\phi,\theta,z,t_0)$. As solution of (1.1) at time t_1, the result of the last step in (1.2) is taken, i.e. $c(\phi,\theta,z,t_1) = c_N(\phi,\theta,z,t_1)$. Applied in this way, the error made in the approximation $c(\phi,\theta,z,t_1)$ is first order in time. If we, however, in the next step from t_1 to t_2, reverse the order of the processes, the error in the solution at t_2 is second order in time. This way of splitting, which is called Strang splitting [59, 40], is applied in EUROS and CWIROS. The integration interval for all processes in (1.2) is equal to half an hour.

The advantage of the operator splitting is clear: since each process is treated separately, for each of the differential equations in (1.2) the most

efficient numerical integration technique can be chosen. For example, in order to solve the chemical equations with sufficient accuracy, a number of time steps with an implicit or semi-implicit solver is required (see Chapter 5 and 6). The advection, however, can be performed using explicit integration techniques with time steps of half an hour. Hence, if advection and chemistry are solved without operator splitting, a (semi-) implicit method would be needed to solve these processes in a coupled way because of the chemistry. This coupling results in large systems of nonlinear equations to be solved, which is computationally very expensive and hence unattractive. Apart from that, in the coupled approach more time steps are taken than necessary for advection alone. Hence, operator splitting is a means to limit the total computation time.

The drawback of operator splitting, however, is the splitting error. It is beyond the scope of this thesis to discuss this issue, but it may very well be that this error is large. In atmospheric models only modest accuracy is required which justifies the application of operator splitting. This thesis, however, focuses on numerical techniques for some of the physical and chemical processes to be applied within the operator splitting context. No further attention is paid to splittings and the associated errors.

1.6 Outline of the thesis

Chapter 2 gives a description of the physical and chemical model. In this chapter all relevant parameters are defined and given a value.

Chapter 3 describes the grid refinement technique. It is argued why finite volumes should be used in CWIROS and a finite volume version of the technique of Trompert and Verwer [63] is presented. Also the application of this technique to the smog model is described.

Chapter 4 describes a few finite volume advection schemes. The emphasis is on positivity (preventing under- and overshoot) and on mass conservation, because these aspects are considered to be important for atmospheric applications. Results of numerical experiments for these schemes are presented.

Chapter 5 discusses special purpose solvers for systems of stiff ODEs arising from chemical kinetics. The emphasis is on efficiency for a modest accuracy requirement. Also attention is paid to mass conservation. In Chapter 6 some of the special purpose solvers are tested together with a state-of-the art solver in a box model test.

Chapter 7 provides a comparison between model results and measurements for a winter and summer smog episode. The comparisons show a good qualitative behavior, but not all details are resolved in the model results.

In the final Chapter 8, conclusions are drawn and recommendations for future research are done.

Chapter 2

Model Description

This chapter gives a description of the model. In earlier stages, model descriptions have been written [17, 39, 47, 48, 49]. In [42] an attempt has been made to write a model description starting from [17, 39, 47, 48, 49], the source code of EUROS and the proposed changes for CWIROS. During the course of the project EUSMOG various changes in the model have been carried through. The present model description therefore is different from the one in [42].

2.1 The model domain

For a medium range smog forecast, a geographical model area of the size of Europe is necessary. The model area is plotted in Fig. 2.1. As can be seen from Fig. 2.1, we do not work with the usual latitude-longitude coordinates. Instead we use a shifted pole coordinate system, i.e. the real North pole is at 30^o Northern latitude in the new coordinate system. In other words, the equator has been shifted to 60^o Northern latitude in the usual latitude-longitude coordinates. A more detailed description can be found in Appendix A. In shifted pole coordinates the model domain is $[-8.25^o, 20.35^o] \times [-23.1^o, 7.15^o]$. Through this choice the smallest mesh widths (expressed in meters) are larger than in the usual coordinate system, when using a uniform grid in latitude-longitude coordinates. In general, this will lead to smaller maximum Courant numbers and thus a less severe restriction on the time step size. The base grid, covering the model area, consists of 52×55 grid cells each representing an area of $.55^o \times .55^o$. This grid system is in agreement with the system used in HIRLAM, a meteorological model developed by various meteorological institutes including the Royal Netherlands Meteorological Institute (KNMI). This model is not operational yet at RIVM, but it is expected to become operational in the future. Its model output then can directly be used as input for CWIROS.

2.1.1 Vertical stratification

The vertical stratification is modeled by 4 layers. In each layer, the concentration is taken vertically homogeneous. The height of the layers varies during the day due to meteorological processes, see Fig. 2.2.

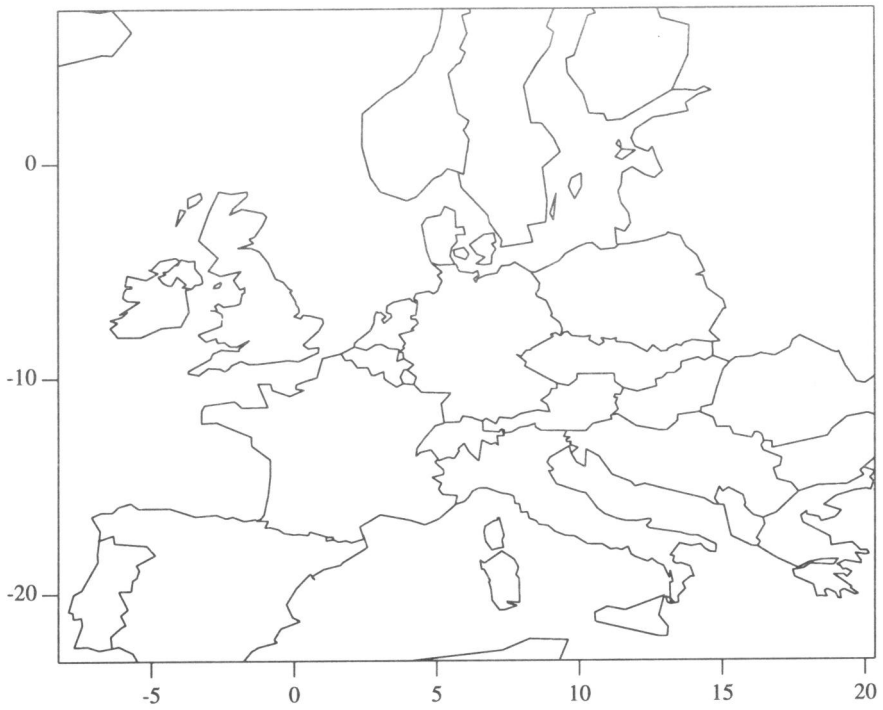


Figure 2.1: The model area

- *Surface layer:* from $z = z_0$ to $z = H_s$. H_s is taken equal to 50m, z_0 is the roughness length, depending on the surface roughness and meteorological parameters (see [75]). In this layer, emissions by traffic and space heating take place. At the surface of the earth, removal of pollutants by dry deposition takes place. Following [48, 49], horizontal advection is neglected in the surface layer. There is only transport of pollutants in vertical direction by vertical (turbulent) diffusion. Whether this assumption is still valid in case of grid refinement, has to be investigated.

- *Mixing layer:* the layer between the top of the surface layer and inversion height z_i. The depth H_m of this layer is constant during the night, grows during morning hours after sunrise and in the late afternoon the nocturnal mixing height is established again very quickly due to sunset. When the mixing height is growing, pollutants from the reservoir layer are injected into the mixing layer. When the mixing height is decreasing, the reverse process takes place. This process is called *fumigation.* A vertically homogeneous concentration is assumed in this layer due to the strong mixing.

- *Reservoir layer:* the layer above the mixing layer with depth H_r. The top of this layer is determined by the effective height of the sources: pollutants emitted above the mixing layer at night are injected in the reser-

voir layer. During the next day, these emissions may be re-entrained into the mixing layer as the mixing height rises. Exchange between the mixing layer and the reservoir layer only occurs by changes of the mixing height. In the afternoon, the reservoir layer may vanish if the mixing height exceeds the top of the reservoir layer.

- *Upper layer:* this layer with depth H_u serves as a 'semi-permanent' reservoir for pollutants released directly from the mixing layer during the afternoon when the reservoir layer has vanished. The height of the top of the upper layer has at least to be equal to the maximum possible mixing height. In the model the top of the upper layer is at $z = 3000$m.

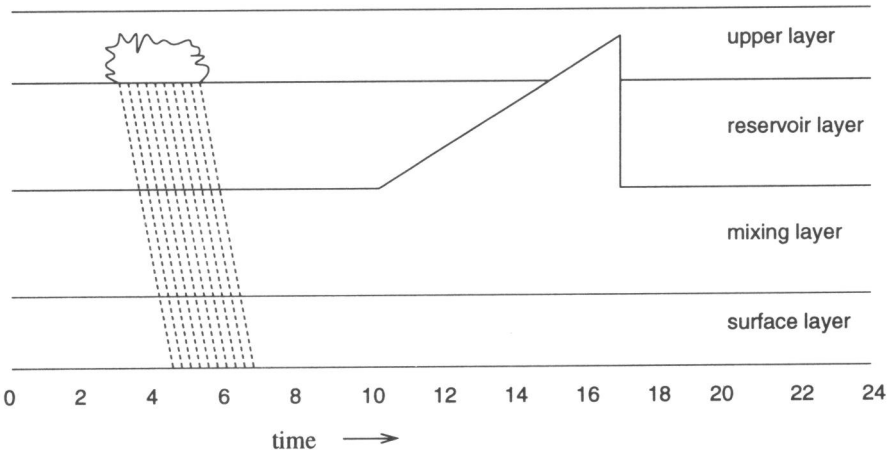


Figure 2.2: The four layers

In the present model the heights of the layers only vary in time but are taken constant over the whole model area. For several reasons this can be somewhat unrealistic. First of all, the height of the mixing layer depends on a lot of meteorological parameters. These parameters cannot be considered to be uniform over the whole model area. Also, the size of the model area is such that there is a time difference of a few hours between sunrise and sunset in the western part and the eastern part of the model area. Furthermore, there will be abrupt changes in the mixing height at land-sea boundaries [49]. However, in smog forecasting for the Netherlands, this is considered not to be very important, because in the Netherlands episodes are generally characterized by continental flows. Improvement could possibly be achieved by using the local mixing height, but in the present model this is not done.

2.2 The physical model

2.2.1 The modeled species

In the model 15 species are taken into account. They are listed below. The numbering corresponds with the numbering used in CWIROS.

1.	SO_2	Sulphur dioxide	2.	SO_4	Sulphate aerosol
3.	NO	Nitrogen oxide	4.	NO_2	Nitrogen dioxide
5.	O_3	Ozone	6.	OH	Hydroxyl radical
7.	NO_3^a	Nitrate aerosol	8.	C_2H_6	Ethane
9.	C_4H_{10}	Butane	10.	C_2H_4	Ethene
11.	C_3H_6	Propene	12.	XYL	Xylene
13.	ISO	Isoprene	14.	CO	Carbon monoxide
15.	HNO_3	Nitric acid			

Methane (CH_4) has also been included, because it is important for ozone formation. Its value has been fixed to 1700 particles per billion (ppb), since for the duration of smog forecasts its reaction time is considered to be too small for a notable change in concentration. Note that the superscript a in NO_3^a means aerosol, which should not be confused with the NO_3 radical. The species 8-14 and methane are called Volatile Organic Compounds (VOCS). These species are essential in ozone formation (the precursors), but do not play any significant role in winter smog prediction.

2.2.2 The model equation

In spherical coordinates, the full model equation can be written as

$$
\begin{aligned}
\frac{\partial c}{\partial t} &= -\frac{1}{r\cos\theta}\left[\frac{\partial(uc)}{\partial\phi} + \frac{\partial(vc\cos\theta)}{\partial\theta}\right] && \text{advection}\\
&+ \frac{\kappa_{diff}}{r^2\cos\theta}\left[\frac{1}{\cos\theta}\frac{\partial^2 c}{\partial\phi^2} + \frac{\partial c}{\partial\theta}\left(\frac{\partial c}{\partial\theta}\cos\theta\right)\right] && \text{horizontal diffusion}\\
&+ \frac{\partial}{\partial z}\left(K_z(z)\frac{\partial c}{\partial z}\right) && \text{vertical diffusion}\\
&+ S_d(c) && \text{dry deposition}\\
&+ S_w(c) && \text{wet deposition}\\
&+ F(c) && \text{fumigation}\\
&+ Q && \text{emission}\\
&+ R(c). && \text{chemical reactions}
\end{aligned}
$$

The ordering in which the processes are treated in the operator splitting, is equal to the order listed above. Dry deposition is included in the vertical diffusion and the emissions are included in the chemical reactions. Note that the vertical diffusion is parametrized in the four layer model, see Section

2.2.5. Since we only have four layers in vertical direction, we work with averaged concentrations in each layer. The concentration in a layer in each horizontal grid cell can be interpreted as an averaged value over this layer in vertical direction. In the remainder of this paper, averaged values will be denoted by a capital C, otherwise a small c is used. For the averaged concentrations and the depths of the layers the subscripts $_{s,m,r,u}$ or $_{1,2,3,4}$ will be used denoting the surface layer, the mixing layer, the reservoir layer and the upper layer, respectively. If there is no danger for confusion, subscripts are omitted. It should also be noted that, strictly speaking, c and C are concentration vectors of length 15. However, apart from chemical reactions, there is no interaction between the components. Therefore, in the description of the various atmospheric processes below, c and C may be interpreted as scalars, unless stated otherwise. In the following a detailed description of the subprocesses in the model equation is given.

2.2.3 Advection

In each of the four layers we have advection. Only for the surface layer, it is not clear yet whether advection can be neglected or not. Advection on a spherical surface can be described by the following equation (see [29, 71])

$$\frac{\partial c}{\partial t} + \frac{1}{r\cos\theta}\left[\frac{\partial(uc)}{\partial\phi} + \frac{\partial(vc\cos\theta)}{\partial\theta}\right] = 0, \tag{2.1}$$

with r the radius of the earth in meters and (ϕ, θ) the coordinates in longitude and latitude direction, respectively. The wind components (in m/s) in longitude and latitude direction are specified by u and v. Note that (2.1) is applied to averaged concentrations C. This can be done if the horizontal wind field is constant in vertical direction within each layer.

As wind fields the 6 or 12-hourly 1000 mbar and 850 mbar wind fields from the European Centre for Medium range Weather Forecasting (ECMWF) in Reading (UK) are available. These wind fields correspond with a certain height depending on the air pressure at ground level. The 1000 mbar field is used for transport in the mixing layer. It is also used for evaluation of the vertical wind profile in the surface layer. Because the ECMWF fields are given on a different grid, spatial interpolation has to be performed. To obtain a wind field at a desired hour, time interpolation will be necessary in general.

To make the wind fields divergence free a routine is included, based on [18]. The main reason to make the wind field divergence free (i.e. the maximum divergence is less than some parameter ϵ) is the fact that the wind component in z direction is omitted. This causes unnatural compression and dilution. In nature wind fields are almost divergence free. In Section 4.5, the procedure for making wind fields divergence free is described in more detail.

2.2.4 Horizontal diffusion

The horizontal diffusion (i.e. in ϕ and θ direction) is given by

$$\frac{\partial c}{\partial t} = \frac{\kappa_{diff}}{r^2 \cos\theta}\left[\frac{1}{\cos\theta}\frac{\partial^2 c}{\partial \phi^2} + \frac{\partial c}{\partial \theta}\left(\frac{\partial c}{\partial \theta}\cos\theta\right)\right], \tag{2.2}$$

where κ_{diff} is the diffusion coefficient. Its value is set equal to $10^4 m^2\, s^{-1}$ on the base grid. On the finer grids (see Chapter 3) κ_{diff} decreases proportional to the square root of the mesh width. The diffusion equation has to be applied for each layer and each component.

2.2.5 Vertical diffusion and dry deposition

The main vertical transport process in the atmosphere can be modeled by the (turbulent) diffusion equation

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial z}\left(K_z(z)\frac{\partial c}{\partial z}\right), \tag{2.3}$$

where K_z is a parametrized diffusion coefficient. This turbulent diffusion process only takes place in the surface and mixing layer. No diffusion occurs in the upper two layers. The same is supposed to be true for dry deposition. However, (2.3) cannot be used directly to describe vertical diffusion in the model, since we only have four layers in the vertical direction. Instead we use (see [39]) the following system of ODEs to describe dry deposition and the exchange between the mixing layer and the surface layer:

$$\frac{\partial C_m H_m}{\partial t} = -\beta v_g(H_s) C_m - \frac{1}{r_a(H_s)}\left(C_m - \frac{v_g(s)}{v_g(H_s)} C_s\right), \tag{2.4a}$$

$$\frac{\partial C_s H_s}{\partial t} = -v_g(s) C_s - \frac{\partial C_m H_m}{\partial t}, \tag{2.4b}$$

where

$$\beta = H_m / (H_m + \frac{v_g(H_s)}{v_g(s)} H_s),$$

$s = \frac{1}{2}H_s$, $r_a(z)$ the aerodynamic resistance and $v_g(z)$ the deposition velocity. In [37] this modeling is discussed in more detail and numerical results for (2.3) and (2.4a,b) are given to support the choice for (2.4a,b). Note that the system (2.4a,b) can be solved exactly.

2.2.6 Wet deposition

Wet deposition is caused by precipitation scavenging. In the upper layer, wet deposition is modeled by (see [19, 39, 49])

$$\frac{\partial C_u}{\partial t} = -\frac{WI}{H_u} C_u,$$

where W is the wash-out coefficient and I the precipitation intensity (m/s). In the layers below the upper layer only partly wash-out takes place. In [39] it is supposed that the wash-out is proportional to the difference between the concentration in the layer itself and the layer above. Also the wash-out

process is assumed to be irreversible, i.e. there is no diffusion of pollutants from the droplets to the air. These assumptions lead to the following equation for the concentrations in the layers below the upper layer,

$$\frac{\partial C_i}{\partial t} = \begin{cases} -\frac{WI}{H_i}(C_i - C_{i+1}) & \text{if } C_i - C_{i+1} > 0 \\ 0 & \text{if } C_i - C_{i+1} < 0 \end{cases} \quad i = 1,2,3 \quad (2.5)$$

where the subscript i denotes the layer number. The wash-out coefficient W depends on the chemical species and meteorological and physical parameters [19, 39]. Values for W are specified in Table 2.1.

The wash-out ratio for SO_2 is dependent on temperature T and pH values.

species		$W\ (m/s)$
SO_2	summer	$5 * 10^4$
	winter	$6 * 10^4$
SO_4		$5 * 10^3 I^{-0.36}$
NO_3^-		$5 * 10^3 I^{-0.36}$

Table 2.1: Values of W for some species

For lower temperatures or higher pH values the wash-out ratio of SO_2 is higher. For this reason a higher wash-out coefficient for SO_2 is taken in winter. Wet deposition for components not listed in Table 2.1 is neglected.

Although the model includes wet deposition, predicted rain fields are not on-line available at RIVM, so only in scenario runs wet deposition can be taken into account. Fortunately, for smog episodes wet deposition plays no significant role.

2.2.7 Fumigation

Fumigation occurs if the mixing height changes. In that case pollutants from the reservoir layer are absorbed by the mixing layer or vice versa. In case the reservoir layer does not exist, exchange of pollutants takes place between the mixing layer and the upper layer. It is easily seen that this process in case of growing mixing height is modeled by

$$\frac{\partial H_m C_m}{\partial t} = \frac{\partial H_m}{\partial t} C_{r,u}. \quad (2.6a)$$

The concentration $C_{r,u}$ denotes the (averaged) concentration in the reservoir layer if this layer exists. Otherwise it denotes the (averaged) concentration in the upper layer. The same holds for $H_{r,u}$, denoting the depth of the reservoir layer H_r or the upper layer H_u. The derivative $\frac{\partial H_m}{\partial t}$ is assumed to have a constant value ζ specified by the meteorological input for the model. In that case (2.6a) can be solved exactly

$$C_m(t+\Delta t) = \frac{H_m(t)}{H_m(t)+\zeta\Delta t}C_m(t) + \frac{\zeta\Delta t}{H_m(t)+\zeta\Delta t}C_{r,u}(t). \quad (2.6b)$$

In the equations (2.6a) and (2.6b) it is assumed that it does not occur that the reservoir layer vanishes and the depth of the upper layer changes due to increasing mixing height in the same time step. If this does occur, it is easily seen that (2.6b) has to be replaced by

$$C_m(t+\Delta t) = \frac{H_m(t)C_m(t) + H_r(t)C_r(t) + (\zeta\Delta t - H_r(t))C_u(t)}{H_m(t)+\zeta\Delta t}.$$

In the afternoon the mixing height drops to its nocturnal value within one time step. In case the reservoir layer does not exist the change in the (averaged) concentrations is given by

$$C_r(t+\Delta t) = C_m(t), \quad (2.7a)$$

$$C_u(t+\Delta t) = \frac{H_u(t)C_u(t)}{H_u(t+\Delta t)} + \frac{H_u(t+\Delta t)-H_u(t)}{H_u(t+\Delta t)}C_m(t), \quad (2.7b)$$

where $H_u(t+\Delta t)$ is a model parameter. In case the reservoir layer already exists, only the concentration in the reservoir layer changes

$$C_r(t+\Delta t) = \frac{H_r(t)}{H_r(t+\Delta t)}C_r(t) + \frac{H_r(t+\Delta t)-H_r(t)}{H_r(t+\Delta t)}C_m(t).$$

It is also possible that no reservoir layer arises. In that case only (2.7b) has to applied.

2.2.8 Emission

Emission is not treated as a separate process in the operator splitting, but it is included in the chemistry. Only yearly averaged emission data are available. From these data, hourly emissions are derived. The way this is done for each layer and each grid cell is outlined below.

Emission categories

Emissions are divided into 6 subcategories:

1. *combustion*
2. *space heating*
3. *refinery*
4. *chemical processes*
5. *solvents*
6. *traffic*

For each of these subcategories yearly averaged emission data specified per source are available. The influence of the month, the day in the week and the hour of day is simulated by 3 parameters,

$\gamma_{m,i}$: for monthly variation,

$\gamma_{d,i}$: for daily variation,

$\gamma_{h,i}$: for hourly variation,

where the subscript i denotes the source category. In the following this subscript is omitted. To obtain the emission at a specific date and hour from a yearly averaged emission $\bar{Q}$ (in ppb/h), we divide $\bar{Q}$ by the number of hours in a year and multiply this by $\gamma_m\gamma_d\gamma_h$. Values for γ_m, γ_d and γ_h can be found in Appendix B. Note that for each source category the relation

$$(12 \times 7 \times 24)^{-1} \sum_{m,d,h} \gamma_m\gamma_d\gamma_h = 1$$

must be satisfied. From the values in Appendix B, it can be seen that this relation holds ($\frac{1}{12}\sum\gamma_m = 1$, $\frac{1}{7}\sum\gamma_d = 1$, $\frac{1}{24}\sum\gamma_h = 1$).

Emitted species

Emission data are available for SO_x (the sum of SO_2 and SO_4), NO_x (the sum of NO and NO_2) and the total VOC. From these data, emissions of SO_2, SO_4, NO, NO_2 can be derived. It is assumed that 3% of the SO_x emission contributes to the concentration of SO_4 while the remaining 97% contributes to the concentration of SO_2. The NO_x emission directly contributes to the NO concentration. Apart from that, 5% of it contributes to the NO_2 concentration. In case the temperature is below 5^oC this percentage is assumed to be 15% in the surface layer due to "cold cars".

The VOC emissions in the database are not specified per component. Therefore a distribution of the VOC emisision over the species is made. The fractions are specified in Table 2.2. The sum of the fractions in Table 2.2 does

C_2H_6	0.0768	C_3H_6	0.0383
C_4H_{10}	0.4144	XYL	0.2454
C_2H_4	0.0364		

Table 2.2: Fractions of the total VOC emission specified per component

not add up to 1. This is because about 20% of the VOC emission is supposed to be in form of species that are not of importance for the smog model and hence not included. Isoprene is emitted by forests. Its emission is dependent on the type of forest (needle or leaves). The fractions for deciduous forest f_d and needle forest f_n is estimated by the formula

$$\begin{aligned} f_d &= 2.2 - 0.032\theta, \\ f_e &= 1 - f_d, \end{aligned}$$

where θ is the latitude in degrees. The emission factor E_d and E_n for both types of forests are estimated by

$$\begin{aligned} E_d &= 242e^{0.06T_c} \quad \mu g/m^3/h, \\ E_e &= 69e^{0.06T_c} \quad \mu g/m^3/h. \end{aligned}$$

where T_c denotes the temperature in degrees Celcius.

Vertical distribution

SO_x and NO_x sources are supposed to emit into the surface layer or into the mixing layer. Within each layer, a homogeneous vertical distribution is assumed. The distribution of the SO_x and NO_x sources between the surface layer and the mixing layer is specified in Table 2.3. Organic components are supposed to be emitted only in the surface layer. The fraction emitted into

source category	SO_x	NO_x
1	0.23	0.23
2	0	0
3	0	0
4	1	1
5	1	1
6	1	1

Table 2.3: Fraction of the emission emitted into the surface layer

the mixing layer partially penetrates into the reservoir layer. This process is investigated and described by Manins [44]. To model this (partial) penetration, two parameters are introduced (see [44]): the fraction f indicating the fraction remaining in the mixing layer, and the penetration parameter P. The fraction f can be expressed in terms of P

$$f = \begin{cases} \max\{0, F(P)\} & \text{if } P > 0.08 \\ 1 & \text{if } P \leq 0.08 \end{cases} \tag{2.8a}$$

with

$$F(P) = 0.08P^{-1} - (P - 0.08). \tag{2.8b}$$

As f is a fraction, its value can never become less than zero or greater than 1. From (2.8a,b) it is easily seen that for all possible values of P we have $0 \leq f \leq 1$. For large values of P, $F(P)$ may become greater than 1, which means total penetration, and in that case f is set equal to zero. If P is less than 0.08, there is no penetration at all, and f is set equal to 1. P is estimated in the following way (see e.g. [44] and [39]):

$$P \equiv \frac{F}{\tilde{u}\, b\, dz^2} \approx \frac{258\,\tilde{Q}}{\Delta T\, \tilde{u}\, dz^2},$$

where

F	=	buoyance flux of the chimney
$\tilde{Q}$	=	heat in MW
b	=	inversion strength
ΔT	=	temperature jump at inversion
dz	=	distance between stack height and inversion
$\tilde{u}$	=	wind velocity at source height

The wind velocity is evaluated using the power law exponent according to (2.10). Based on statistics on sources in the Ruhr area, a homogeneous distribution of the sources is assumed between 25 and $150m$. Further, it is assumed that the heat of the sources is a function of the height h defined by

$$\tilde{Q} = \text{const}\, h^3, \quad \text{const} = 1.6\,10^{-5}\ \text{MW}/m^3.$$

The temperature jump ΔT is specified as a function of g_{min} and g_{max} by

$$\Delta T = g_{max} - \frac{(g_{max} - g_{min})\, f_{tim}}{t_{sr} - t_{sf}},,$$

where f_{tim} is the time passed since the start of the fumigation. If however $f_{tim} > t_{sr} - t_{sf}$ or $f_{tim} < 0$, f_{tim} is put to zero. So ΔT is supposed to be a (piecewise) linear function in time. To calculate the total emission into the mixing layer and the reservoir layer, we have to integrate $f(z)Q(z)$ over the mixing layer

$$Q_m = \int_{H_s}^{H_s+H_m} f(z)Q(z)dz. \tag{2.9}$$

Because a vertically homogeneous distribution of the emission is assumed, $Q(z)$ in (2.9) may be replaced by $\bar{Q}/H_m$, yielding

$$Q_m = \frac{\bar{Q}}{H_m} \int_{H_s}^{H_s+H_m} f(z)dz.$$

2.3 The chemical model

The chemistry in the model consists of an ozone model, proposed to us by F. de Leeuw [38]. This model has been derived from the EMEP MSC-West model [57] and is highly parametrized. The EMEP model consists of about 140 reactions between ca. 70 species. However, as the model has to be applicable to a large model area and all meteorological conditions, an attempt is made to retain the essential characteristics of photochemical ozone formation. The photochemical generated ozone is calculated according (see Stedman and Williams [58] and references cited therein):

$$\frac{d[O_3]}{dt} = \sum a_i k_i [OH][VOC_i],$$

where VOC_i is an organic compound, k_i is the reaction constant for the initial reaction of VOC_i with the OH radical. Further, a_i is the stoichiometry factor for VOC_i, that is the total number of ozone molecules generated by complete degradation of one VOC molecule. Although a_i depends on a number of factors, here only a dependence on NO_x concentrations is assumed, that limits the ozone production at low NO_x concentrations. The following reactions with their reaction constants k_i are taken into account (see [57, 73])

1.	$NO + O_3$	$\longrightarrow$	NO_2	k_1	$= 1.0\ 10^{-12} \exp(-1400/T_k)$
2.	$NO_2 + h\nu$	$\longrightarrow$	$NO + O_3$	k_2	$= 1.45\ 10^{-2} \exp(-0.4/\cos\gamma)$
3.	$NO_2 + OH$	$\longrightarrow$	HNO_3	k_3	$= 1.68\ 10^{-12} \exp(560/T_k)$
4.	$2NO_2 + O_3$	$\longrightarrow$	$2NO_3^a$	k_4	= see below
5.	$O_3 + h\nu$	$\longrightarrow$	$b_1 OH + b_2 O_3$	k_5	$= 2.0\ 10^{-4} \exp(-1.4/\cos\gamma)$
6.	$C_2H_6 + OH$	$\longrightarrow$	$a_1 O_3$	k_6	$= 8.7\ 10^{-12} \exp(-1070/T_k)$
7.	$C_4H_{10} + OH$	$\longrightarrow$	$a_2 O_3$	k_7	$= 1.4\ 10^{-11} \exp(-559/T_k)$
8.	$C_2H_4 + OH$	$\longrightarrow$	$a_3 O_3$	k_8	$= 1.66\ 10^{-12} \exp(474/T_k)$
9.	$C_3H_6 + OH$	$\longrightarrow$	$a_4 O_3$	k_9	$= 4.1\ 10^{-12} \exp(545/T_k)$
10.	XYL + OH	$\longrightarrow$	$a_5 O_3$	k_{10}	$= 1.4\ 10^{-11}$
11.	ISO + OH	$\longrightarrow$	$a_6 O_3$	k_{11}	$= 2.55\ 10^{-11} \exp(410/T_k)$
12.	$CO + OH$	$\longrightarrow$	$a_7 O_3$	k_{12}	$= 2.4\ 10^{-13}$
13.	$CH_4 + OH$	$\longrightarrow$	$a_8 O_3$	k_{13}	$= 2.9\ 10^{-12} \exp(-1820/T_k)$
14.	$SO_2 + OH$	$\longrightarrow$	SO_4	k_{14}	$= 2.32\ 10^{-10} \exp(-457/T_k)$
15.	SO_2	$\longrightarrow$	SO_4	k_{15}	$= 1.39\ 10^{-6}$
					$= 4.17\ 10^{-6}$ in upper layer
16.	HNO_3	$\longrightarrow$	$NO_2 + OH$	k_{16}	$= 3.0\ 10^{-6} \exp(-1.25/\cos\gamma)$
17.	HNO_3	$\longrightarrow$	NO_3^a	k_{17}	$= 5\ 10^{-6}$

- The concentrations are specified in number of molecules (mlc) per cubic centimeter, mlc/cm^3. An often used unit of measure for concentrations is ppb (particles per billion). If we assume that 1 mol (= 6.022 10^{23} molecules) has a volume of 24.4 liter, 1 ppb corresponds with $2.46\,10^{10}\ mlc/cm^3$.

- The parameters b_1 and b_2 are given by

$$\begin{aligned} b_1 &= \frac{2k_A\,[H_2O]}{k_A\,[H_2O] + k_B} \\ k_A &= 2.3\ 10^{-10} \\ k_B &= 4.93_{10}{}^{8} \exp(-100/T_k) \\ b_2 &= 1 - \frac{b_1}{2} \end{aligned}$$

  where T_k denotes the temperature in degrees Kelvin.

- The water concentration in the air is dependent on meteo conditions and is calculated according to [45]. The water concentration $[H_2O]$ (in

mlc/cm^3) is supposed to be given by

$$[H_2O] = \frac{4.357521\,10^{19}\,rh}{T_k} \times \exp\left(-(753.0 - 0.57T_k)(\frac{1}{T_k} - \frac{1}{273.16})18/1.986\right),$$

where rh is the relative humidity ($0 \leq rh \leq 1$).

- The parameters a_i are functions of the NO_x concentration (the sum of NO and NO_2) according to

$$a_i = a_{i,max}\, e^{-b_i/[NO_x]} + a_{i,min} \quad \text{(with } [NO_x] \text{ in ppb)}.$$

The following (preliminary) values are used for the $a_{i,max}$ and $a_{i,min}$:

i	$a_{i,max}$	$a_{i,min}$	b_i	i	$a_{i,max}$	$a_{i,min}$	b_i
1	4.4	1.7	0.35	5	7.0	3.0	0.35
2	5.8	2.2	0.35	6	0.0	0.1	0.35
3	5.5	2.2	$5\,10^{-4}$	7	0.9	0.0	0.25
4	7.0	0.4	$5\,10^{-4}$	8	4.0	0.0	0.25

- Reaction 4 is a net reaction obtained by lumping of reactions with the NO_3 radical (which should not be confused with its aerosol NO_3^a) and N_2O_5:

a.	$NO_2 + O_3$	$\longrightarrow$	NO_3	k_a	$= 1.2\ 10^{-13} \exp(-2450/T_k)$
b.	$NO_3 + NO$	$\longrightarrow$	$2NO_2$	k_b	$= 1.5\ 10^{-11} \exp(170/T_k)$
c1.	$NO_3 + h\nu$	$\longrightarrow$	NO_2	k_{c1}	$= 0.192 \exp(-0.059/\cos\gamma)$
c2.	$NO_3 + h\nu$	$\longrightarrow$	NO	k_{c2}	$= 0.0243 \exp(-0.081/\cos\gamma)$
d.	$NO_3 + NO_2$	$\longrightarrow$	N_2O_5	k_d	$= 1.47\ 10^{-12} \exp(-60/T_k)$
e.	N_2O_5	$\longrightarrow$	$NO_2 + NO_3$	k_e	$= 8.5\ 10^{14} \exp(-11080/T_k)$
f.	$N_2O_5 + H_2O$	$\longrightarrow$	$2HNO_3$	k_f	$= 1.3\ 10^{-21}$
g.	HNO_3	$\longrightarrow$	NO_3^a	k_g	fast

Neglecting the intermediate product HNO_3, reaction f and g can be rewritten into a single reaction f':

$$f'.\quad N_2O_5 + H_2O \longrightarrow 2NO_3^a \quad k_{f'} = k_f$$

The concentrations of NO_3 and N_2O_5 are assumed to be in steady state, i.e.

$$\frac{\partial}{\partial t}[N_2O_5] = 0$$

and

$$\frac{\partial}{\partial t}[NO_3] = 0.$$

This leads to

$$[N_2O_5] = \frac{k_d[NO_2]\,[NO_3]}{k_e + k_f[H_2O]}$$

and

$$[NO_3] = \frac{k_a[NO_2][O_3]}{k_b[NO] + k_c + (k_d k_f[NO_2][H_2O])/(h_e + k_f[H_2O])},$$

with $k_c = k_{c1} + k_{c2}$. The concentrations of NO_3 and N_2O_5 are thus supposed to be functions of other concentrations and reaction rates. To obtain the net reaction 4, consider f'. For the production of NO_3^a, N_2O_5 is needed. This is formed by reaction d which in return needs NO_3, produced by reaction a. Combining a and d gives

$$2NO_2 + O_3 \longrightarrow 2NO_3^a.$$

The loss of of N_2O_5, and thus of NO_2 and O_3, is described by reaction f'. So the effective reaction constant k_{eff} of reaction 4 can be expressed in terms of k_f, N_2O_5 and H_2O:

$$k_{eff} = k_f \frac{[N_2O_5]\,[H_2O]}{[NO_2]^2\,[O_3]}.$$

- The reaction constants of some of the reactions depend on temperature and solar angle. The solar angle γ is dependent on the time of the year, the time of day t (in hours) and the longitude and latitude coordinates (ϕ, θ) of the point considered. The cosine of the solar angle is given by

$$\cos\gamma = \sin\Delta\sin\theta + \cos\Delta\cos\theta\cos(\frac{\pi}{12}(t - 12.67)),$$

where

$$\begin{aligned}\Delta = {} & 0.006918 - 0.399912\cos\tilde{d} + 0.070257\sin\tilde{d} - \\ & 0.006758\cos(2\tilde{d}) + 0.000907\sin(2\tilde{d}) - \\ & 0.002697\cos(3\tilde{d}) + 0.00148\sin(3\tilde{d})\end{aligned}$$

with

$$\tilde{d} = \frac{2\pi d}{365}$$

and d the day in the year ($1 \leq d \leq 365$).

- All reaction constants depending on $\cos\gamma$ are multiplied by a factor depending on the cloud coverage parameter N

$$1 - 0.75N^{3.4}.$$

2.4 Meteorological parameters

2.4.1 Input parameters

As CWIROS has to be an operational code, its input has to be easily available, without carrying out special measurements. The input parameters for CWIROS are listed in Table 2.4. In case no input parameters are available, the code uses default values, also listed in Table 2.4. Other parameters are

para-meter	unit	default		meaning
		wint.	summ.	
u, v	m/s	-	-	wind fields in coordinate directions (ϕ, θ)
$\partial h/\partial t$	m/h	50	75	growth of mixing height during fumigation
mlayer	m	300	300	mixing height during the night
t_{sf}	h	9.00	8.00	starting time of fumigation
t_{ef}	h	12.00	11.00	time at which reservoir layer vanishes
t_{sr}	h	17.00	18.00	time of afternoon stratification
N	-	.5	0.01	cloud coverage fraction
T	^oC	12	20	temperature in degrees Celsius
albedo	-	0.2	0.2	fraction of directly reflected solar radiation
α		0.95	0.95	modified Priestly-Taylor parameter

Table 2.4: Meteorological input parameters

given a constant value within the code:

z_0:	$0.25m$	roughness length (over land)
	$10^{-4}m$	roughness length (over sea)
z_{ra}:	$10m$	height at which the deposition velocities are evaluated
g_{min}:	1	minimum temperature gradient
g_{max}:	10	maximum temperature gradient

From these parameters other parameters are derived.

2.4.2 Stability and deposition parameters

In Table 2.5 stability and depostion parameters are listed which depend on the input parameters. In this section we will describe their derivation. The power law exponent $p > 0$ is used to evaluate the vertical wind profile in the surface layer, which is supposed to be given by

$$\tilde{u}(z) = \tilde{u}(z_{ref}) \left[\frac{z}{z_{ref}} \right]^p, \quad z_{ref} = 250m. \tag{2.10}$$

parameter	unit	meaning
p	-	power law exponent for wind profile
L	m	Monin-Obukhov length (stability parameter)
u_*	m/s	friction velocity
$K_z(z)$	m^2/s	vertical diffusion coefficient
$r_a(z)$	s/m	aerodynamic resistance
r_s	s/m	laminair boundary layer resistance

Table 2.5: Parameters depending on meteorological input

where $\tilde{u}$ denotes the wind velocity (i.e. $\sqrt{u^2+v^2}$). The wind velocity at $z = z_{ref}$ is supposed to be known and specified by the 1000 mbar wind field. This corresponds with an air pressure of about 1033 mbar at sea level, which is a realistic value for smog episodes. The power law exponent is taken equal to

$$p = \begin{cases} 0.10 & \text{if } \frac{1}{L} < -0.55, \\ 0.16 & \text{if } -0.55 \leq \frac{1}{L} < 0.17, \\ 0.30 & \text{if } \frac{1}{L} \geq 0.17. \end{cases} \tag{2.11}$$

Another representation of the vertical wind profile is given by [39]

$$\tilde{u}(z) = \frac{u_*}{k}\left[\ln\left(\frac{z}{z_0}\right) - \psi\left(\frac{z}{L}\right)\right], \tag{2.12}$$

where k is the Von Karmann constant (≈ 0.40) and ψ the stability function. For neutral and unstable conditions ($L \leq 0$), ψ is specified by

$$\psi\left(\frac{z}{L}\right) = 2\ln\frac{1+x}{2} + \ln\frac{1+x^2}{2} - 2\arctan x + \frac{\pi}{2}, \tag{2.13}$$

with $x = (1-16\frac{z}{L})^{\frac{1}{4}}$, and for stable conditions ($L > 0$) by

$$\psi\left(\frac{z}{L}\right) = -5.2\frac{z}{L}.$$

The Monin-Obukhov length L is defined by

$$L = -\frac{\rho C_p T_k u_*^3}{k g H_0}, \tag{2.14}$$

where $\rho = 1.2754$, the density of air in kg/m^3, $C_p = 1.005$, the specific heat, T_k the temperature in degrees Kelvin, and g the gravity constant ($\approx 9.8 m/s^2$). H_0 is the sensible heat flux (kW/m^2). For its parameterization and the way p, u_* and L are calculated using the relations given by (2.10)-(2.14), we refer to [3, 13]. Over sea we always assume $L \gg 0$. In that case we have $p = 0.16$ and $\psi(\frac{z}{L}) \approx 0$. Combining (2.10) and (2.12) evaluated in $z = z_{ra}$ then gives

$$u_* = \frac{\tilde{u}(z_{ref})}{\ln(z_{ra}) - \ln(z_{ref})} \left[\frac{z_{ra}}{z_{ref}}\right]^{0.16}.$$

From these parameters, the vertical diffusion coefficient $K_z(z)$, the aerodynamic and boundary layer resistances, r_a and r_s, can be determined. $K_z(z)$ is supposed to be approximated by (see [37])

$$K_z(z) = \frac{k\, u_*\, z}{\Phi_h}, \tag{2.15}$$

with Φ_h the stability correction function given by

$$\Phi_h = \begin{cases} 0.74\sqrt{(1 - 9\frac{z}{L})} & \text{if } L < 0, \\ 0.74 + 4.7\frac{z}{L} & \text{if } L > 0. \end{cases}$$

The aerodynamic resistance r_a is defined as

$$r_a(z) = \int_{z_0}^{z} [K(z)]^{-1}\, dz,$$

which yields, using (2.15),

$$\begin{aligned} r_a(z) &= \frac{0.74}{k\, u_*}\left(\ln(\frac{z}{z_0}) + \chi(z_0) - \chi(z)\right), \\ \chi(z) &= \begin{cases} -6.4\frac{z}{L} & \text{if } L > 0, \\ 2\ln(y(z) + 1) & \text{if } L < 0, \end{cases} \\ y(z) &= \sqrt{1 - 9\frac{z}{L}}. \end{aligned} \tag{2.16}$$

Species		r_c sea	r_c land
	summer	10	70
SO_2	winter	10	100
	snow covered/frozen underground	10	500
NO_2		∞	200
NO		∞	600
O_3		∞	100
HNO_3		10^{-5}	10^{-5}

Table 2.6: Values for the surface resistance r_c (s/m)

If a different approximation is used for K_z, then the aerodynamic resistance will be different. However, r_a needs only to be evaluated at relatively low heights ($z \ll z_i$, with z_i the mixing height) where (2.15) is a good approximation for K_z. The boundary layer resistance is modeled as

$$r_s = \frac{2.6}{k\,u_*} \,. \tag{2.17}$$

The total resistance is given by the sum of the three resistances $r_a + r_s + r_c$. The deposition velocity is the reciprocal of the total resistance

$$v_g(z) = [r_a(z) + r_s + r_c]^{-1} \,. \tag{2.18}$$

Values for r_c are given in Table 2.4.2. Species not listed in Table 2.4.2 are assumed to have $r_c = \infty$ both over land and over sea. Note that $r_c = \infty$ implies that no deposition takes place.

Chapter 3

Local Uniform Grid Refinement

3.1 Introduction

In Chapter 2 the physical aspects of the Dutch Smog Prediction Model, including the choice of the base grid, have been described. As base grid we have a uniform grid with mesh widths of $.55^o$ in both horizontal directions. This means that the physical grid distances are about 61 km in North-South direction, and vary between 56 km and 61 km in East-West direction. This grid is too coarse to represent local phenomena well enough. For example, the enhancement in concentrations in urban areas, resulting from local emissions, will not be resolved in the present grid. In general, point source emissions will directly be smeared out over a single grid cell, introducing an unnatural amount of diffusion into the model. To represent such local phenomena, a much finer grid is necessary. However, a uniform grid with mesh widths of, say, 10 km would already require more than 30 times as much grid cells as we have presently on the base grid and the computation time would increase with approximately the same factor. For routine smog predictions, this would be too expensive in terms of computation time, as the model calculations have to be done within a few hours on a workstation. Yet we need more resolution to represent local phenomena more precisely. This leads us to the concept of local grid refinement. The basic idea behind this technique is that a higher resolution is only needed in certain areas of the model domain, for example near (point) sources and strong gradients in the concentration field. In these areas the grid is refined and a more accurate solution is obtained. In other parts of the model domain the solution on the base grid will be good enough and no refinement is necessary there. By refining the grid only in areas where it seems necessary, much less grid cells are needed to obtain a solution comparable to the solution obtained by using the fine grid on the whole model domain. This requires that the areas in the model domain where the grid will be refined can be determined in a dynamic way. In other words, it should be possible to create a new refined grid every time step again. The reason for this is that the need for more resolution is not restricted to the direct surroundings of the location of the sources. Generally speaking, refinement will be necessary in areas with large solution gradients. Large gradients are likely to occur not only in areas with strong emissions but also (under certain circumstances) in other areas downwind from sources. This will certainly be

the case during smog episodes.

The aim of this chapter is to describe the grid refinement technique and its application to the Dutch Smog Prediction Model. First, the choice for a specific method is described. Next, a general description of the refinement technique is given, followed by the actual application to the four layer model.

3.2 Choice of method

When choosing a grid refinement method one should take into account the specific application the method will be used for. In the smog model processes are present that satisfy mass conservation relations, for example, horizontal advection and diffusion and also emission. For these processes numerical schemes are selected that conserve mass as well. Therefore, it is natural to require that the grid refinement technique does not disturb the conservation of any conservative integration scheme, or disturbs it only to a very limited extent. A grid point approach does not satisfy this requirement, as we will show later, and therefore we choose the finite-volume approach. This choice is in accordance with the physical model which is explicitly in terms of vertically averaged concentrations, but also in horizontal direction due to the way emissions have to be modeled.

Because the smog model is a four layer model in which vertical processes are parametrized, grid refinement in vertical direction makes no sense. For this reason only grid refinement in horizontal direction is applied. Moreover, in each layer the same grid structure will be used, thus avoiding complicated (and therefore expensive) treatment of vertical processes due to different grid structures. In the description of the grid refinement technique it therefore suffices to consider two-dimensional problems with one component only. Extension to four layers with a number of components is straightforward and will only briefly be discussed.

As we chose the finite-volume approach, the grid refinement techniques developed by Trompert & Verwer [63, 64], Arney and Flaherty [1], Berger and Oliger [5], Gropp [21, 22, 23] in their present form are not suited for our purpose. Yet, these methods are of interest because they can easily be adapted for the finite-volume context. We will not discuss the differences between the various methods as, in our opinion, these methods are comparable and mainly differ in the way the datastructure is built up and consequently how grid structures may look like. A finite-volume grid refinement algorithm is presented by Berger and Colella [4]. The basic idea of this method is the same as of the others, only their choice of datastructure is heavily dominated by practical considerations and the actual application. As a consequence, each grid level consists of a set of rectangular subgrids, possibly overlapping. The reason to require (sub)grids to be rectangular is to avoid (user) implementation of complicated numerical algorithms for a specific grid structure and loss of computational efficiency, because otherwise the algorithms have to be applied on irregular meshes. The consequence of this requirement is an increased overhead, due to the complicated way the grid structures are

created and, in addition, extra computations in overlapping grid cells. We do not follow this approach for two reasons. First of all, a number of processes in the smog model [42] has no horizontal coupling and just a simple loop over all grid cells is sufficient. Secondly, even if there is horizontal coupling (e.g. in advection and diffusion) it depends on the specific datastructure and numerical scheme whether or not loss of computational efficiency will occur. In view of our experience with the method of Trompert & Verwer [63, 64], we do not, generally speaking, expect a significant loss of computational efficiency. We therefore adopted their basic ideas and used them to construct our own finite-volume grid refinement algorithm. The datastructure however has been adapted for specific use in the smog model. Also a few changes had to made because of the formulation in spherical instead of Cartesian coordinates. The actual implementation of the datastructure is based on a code written by Blom, described in [6], in which the changes in the datastructure are carried through.

3.3 Local Uniform Grid Refinement

In this section the local uniform grid refinement technique will be described. First a global outline is given and next each step of the refinement procedure will be discussed in more detail. The datastructure will not be discussed here. The interested reader is referred to Appendix C where a description of the datastructure is given.

3.3.1 Algorithmic Outline

The first step of the algorithm is integration from time level T to $T + \Delta T$ on the base grid. Next, the algorithm checks whether grid refinement is necessary. If so, a new, fine grid is created and the integration from T to $T + \Delta T$ is redone on this grid, if necessary in more than one time step. Missing initial values on the fine grid are obtained by interpolation. After integration on the fine grid, the algorithm checks whether further refinement is necessary and if so, a second level of refinement is created which is treated in the same way as the first level. This process of creating even finer grids is continued until the solution on a certain level becomes acceptable or until a prescribed maximum number of levels is reached. At this point the time integration step is finished. Now the solution on each grid level is injected into the solution on the next lower grid level. This means that the solution values in grid cells of the fine grid are used to obtain solution values in the grid cells of the underlying coarse grid. This has to be done in a top down manner, of course, because the solution at the finest grid is expected to be the most accurate solution. One step of the time integration consists of the following steps:

```
1.  integrate on base grid
2.  check if and where refinement is necessary
3.  if no (further) refinement is necessary: goto 9
```

```
4.   create datastructure of new grid
5.   determine initial values on new grid
6.   determine boundary conditions
7.   integrate on new grid
8.   if #level < max_level goto 2
9.   inject solution
```

3.3.2 How to refine?

Suppose we have integrated on the base grid from T to $T + \Delta T$. Using some criterion (to be specified later), the algorithm decides in which grid cells the solution is not good enough. If according to the criterion a certain cell needs to be refined, the eight cells directly surrounding this cell will also be refined. In this way a safety buffer is created that will prove useful later on. Boundary cells may not have eight such cells. In that case only the existing ones are refined. A cell is refined by bisecting all sides, see Fig. 3.1. This refinement procedure implies that one flagged cell already results into at least a 4×4 grid on the next grid level in case of a boundary cell, and into a 6×6 grid in case of an interior cell. This implies that numerical methods with stencils of four or less points (cells) in one direction can be applied in a consistent way. Note that this procedure allows the refined grid to consist of several disjunct

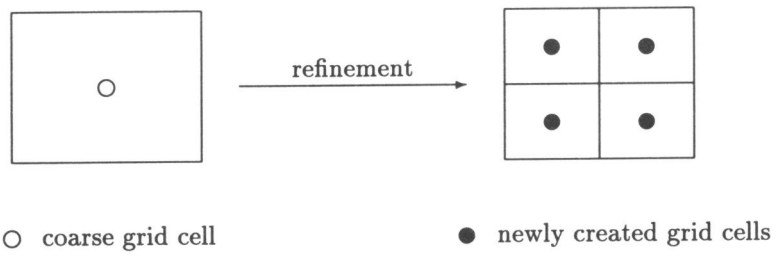


Figure 3.1: Refinement of a grid cell

subgrids, see for example Fig. 3.2.

3.3.3 Where to refine?

Suppose we have integrated on a certain grid level l and we have obtained a solution c^l on this level. To determine where refinement is necessary, some error estimate is needed. We use the curvature of the solution as error indicator, similar as in [63], based upon the second order derivatives of the solution. Since we only need an expression for the curvature of the solution with respect to the coordinate system, or rather the computational grid, we simply use the expression

$$(\Delta\phi)^2|c^l_{\phi\phi}| + (\Delta\theta)^2|c^l_{\theta\theta}|, \tag{3.1}$$

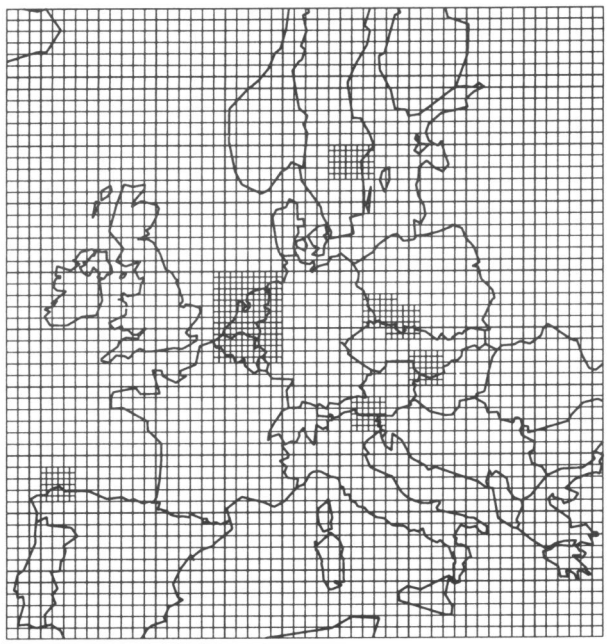


Figure 3.2: Example of grid refinement with two grid levels

which is approximated by standard second order central finite differences in internal grid cells and by uncentered first order differences in boundary cells. The longitude and latitude coordinates are denoted by ϕ and θ respectively. A grid cell is flagged if in this cell

$$(\Delta\phi)^2|c^l_{\phi\phi}| \; + \; (\Delta\theta)^2|c^l_{\theta\theta}| > tol \, \min(c^{max}, c^l_{max}), \tag{3.2}$$

where *tol* is a tolerance value specified by the user, c^l_{max} is the maximum value of c^l at time $T + \Delta T$, and $c^{max} > 0$ is a user defined value. The factor $\min(c^{max}, c^l_{max})$ in the right hand side of (3.2) is added in order to make the refinement criterion independent of the scaling of the problem. The reason why the minimum of the maximum value of the solution and the parameter c^{max} is taken, is that due to emissions c^l_{max} may become so excessively high, especially when $l > 1$, that refinement outside the emission area(s) would not occur otherwise. This consideration suggests a value for c^{max} in the order of magnitude of the natural background value for species c.

In addition to the refinement criterion (3.2), the user has the possibility to enforce refinement in certain areas. To achieve this, the code not only flags cells satisfying (3.2) but also flags cells indicated by the user. These cells may correspond to certain longitude-latitude coordinates, specifying an area of special interest, for example the Netherlands, or to certain emission areas if not already flagged due to (3.2). In the same way the user might

unflag cells in areas in which he does not want the code to create refined grids, albeit that this has to be done very carefully and with good knowledge of the physical model.

Recall that this section describes the grid refinement procedure as if there exists only one layer and one component. In Section 3.4 the application of the grid refinement to the four layer model with 15 components. In Section 3.4.3 the equivalent of criterion (3.2) will be specified for the full model.

3.3.4 Interpolation of initial values

If a fine grid level l has been created, initial values are needed for the start of the integration. Three different situations are distinguished:

- If it is the initial time step, the initial values on each grid level are supposed to be specified by initial conditions. A subroutine is available that assigns initial values to grid cells.
- If level l also existed in the previous time step, values in coinciding cells are copied.
- For all other cells interpolation has to be carried out.

If interpolation has to be carried out, we require that the interpolation procedure is mass conserving, i.e. the sum of the mass in the four fine grid cells should be equal to the mass in the underlying coarse grid cell. This is evidently the case when using constant interpolation. For higher order interpolation methods, it is more complicated to achieve conservation of mass. Suppose a coarse grid cell with value c_0 is refined and the values in the fine grid cells need to be obtained from interpolation. Let us denote them by c_1, c_2, c_3, c_4. The coordinates of the cell centers are given by (ϕ_i, θ_i), with $i = 0, \ldots, 4$. In spherical coordinates, the mass in a grid cell $[\phi_i - \frac{1}{2}\Delta\phi, \phi_i + \frac{1}{2}\Delta\phi] \times [\theta_i - \frac{1}{2}\Delta\theta, \theta_i + \frac{1}{2}\Delta\theta]$ is given by (taking c_i constant or as averaged concentration over the cell) $c_i S_i$ where S_i is the surface integral over the cell defined by

$$S_i(\Delta\phi, \Delta\theta) = \int_{\phi_i - \frac{1}{2}\Delta\phi}^{\phi_i + \frac{1}{2}\Delta\phi} \int_{\theta_i - \frac{1}{2}\Delta\theta}^{\theta_i + \frac{1}{2}\Delta\theta} \cos\bar{\theta} d\bar{\theta} d\bar{\phi} = \Delta\phi\Delta\theta \cos\theta_i \frac{\sin(\frac{1}{2}\Delta\theta)}{\frac{1}{2}\Delta\theta}. \quad (3.3)$$

From expression (3.3) it can be seen that the surface integral is not only proportional to $\Delta\phi\Delta\theta$, as would be expected from the corresponding expression in Cartesian coordinates, but depends also on the latitude and the actual mesh size in latitude direction. Imposing a mass balance for the coarse grid cell gives the condition for mass conserving interpolation

$$\sum_{i=1}^{4} c_i S_i(\frac{1}{2}\Delta\phi, \frac{1}{2}\Delta\theta) = c_0 \sum_{i=1}^{4} S_i(\frac{1}{2}\Delta\phi, \frac{1}{2}\Delta\theta), \quad (3.4)$$

which can be simplified to

$$\sum_{i=1}^{4}(c_i - c_0)\cos\theta_i \;=\; 0. \tag{3.5}$$

The condition for mass conserving interpolation (3.5) suggests interpolation of the function $c(\theta)\cos\theta$ in order to obtain values c_i, $i = 1, \ldots, 4$. Condition (3.5) is equivalent with

$$\sum_{i=1}^{4} c_i \cos\theta_i \;=\; 4c_0 \cos\theta_0 \cos(\frac{1}{4}\Delta\theta). \tag{3.6}$$

If the function $\tilde{c}{=}c\cos\theta$ is used in the following way to obtain interpolated values c_i

$$c_i \cos\theta_i \;=\; c_0 \cos\theta_0 \;+\; \tilde{c}_\theta(\theta_i - \theta_0) \;+\; \tilde{c}_\phi(\phi_i - \phi_0) + \text{ h.o.t.} \tag{3.7}$$

with $\tilde{c}_\phi$ and $\tilde{c}_\theta$ approximations for the derivatives of the function $\tilde{c}$ in (ϕ_0, θ_0), we obtain

$$\sum_{i=1}^{4} c_i \cos\theta_i \;=\; 4c_0 \cos\theta_0. \tag{3.8}$$

Hence, the conservation condition (3.6) is slightly violated. Exact conservation can be imposed by multiplying the interpolated values c_i by a factor $\cos(\frac{1}{4}\Delta\theta)$. For the base grid, this factor is already larger than 0.99999 so that omitting this factor will hardly be felt.

In the model however, we simply use constant interpolation. In practise, interpolation of initial values will only be necessary near the boundaries of newly created grid levels where the solution is to be expected relatively smooth. Recall that the refinement procedure causes a safety buffer to be created by refining cells that need not to be refined based on the error estimate only.

3.3.5 Injection of solution values

When the time integration step is finished, either because the algorithm decided that no further refinement is necessary, or because a prescribed maximum number of grid levels has been reached, the solution values of the fine grid cells have to be injected into the solution values of the underlying coarse grid cells. Again the requirement is that this has to be done in a mass conserving way. The obvious way to perform the injection is to sum up the mass for each set of four fine grid cells that form one coarse grid cell and assign this mass to the coarse grid cell. In terms of average concentrations over a grid cell, mass conserving injection of four fine grid values to one coarse grid value, using the notation of the previous subsection, can be written as

$$c_0 \;=\; \frac{1}{4}\sum_{i=1}^{4} \frac{\cos\theta_i}{\cos\theta_0 \cos(\frac{1}{4}\Delta\theta)} c_i. \tag{3.9}$$

Again we note that omitting the factor $\cos(\frac{1}{4}\Delta\theta)$ will hardly be felt. However, this factor has to be computed only once per injection step, so it does not require much extra computation time to take this factor into account. The same holds for $\cos\theta_i$, $i = 0, \ldots, 4$, since an array containing the cosines of all cell center coordinates is stored if a new grid is created, because they are also needed in other parts of the model computation. Therefore, there is no reason to apply a different formula than (3.9).

It is the injection step in which a grid point approach fails to be conservative. In such an approach, a quarter (ignoring boundaries) of the fine grid point coincides with coarse grid points. Injection of solution values then is straightforward. Values in fine grid points coinciding with coarse grid points are copied, other values in fine grid points do not influence the solution on the coarse grid. Though formally we cannot speak about mass on a grid in the grid point approach, we define the mass on a grid with mesh widths (for simplicity in Cartesian coordinates) Δx and Δy as the sum of all solution values times $\Delta x \Delta y$, being a second order approximation of the integral of the concentration function over the domain. It is now easy to see that a different mass is injected from the fine grid than the mass present on the fine grid. The mass on the fine grid M_F is given by the sum of the concentration values times $\frac{1}{4}\Delta x \Delta y$. The injected mass M_I, however, is equal to the sum of the concentration values in grid points coinciding with coarse grid points times $\Delta x \Delta y$. In general, these two sums are not equal. A simple example clearly shows the inconsistency of the grid point approach at this point. Consider a point source on the fine grid. The only possible way to treat a point source is to assign the emitted mass M_E to the nearest grid point. If this grid point is not coinciding with a coarse grid point, the mass M_E is not injected into the coarse grid solution and the emission is lost on the coarse grid. On the other hand, if this grid point does coincide with a coarse grid point, the solution value is copied. In that case the increase of mass due to the emission of M_E is equal to $4M_E$, because the grid point on the coarse grid represents a four times larger area.

The above considerations show that a grid point approach is not suited for this application. The present finite-volume approach deals with mass conservation in a quite natural and consistent way, and in our opinion this approach is to be preferred in atmospheric models.

3.3.6 Boundary conditions

Assuming boundary conditions are prescribed for cells that abut the physical boundary, no problems arise when integrating on the base grid. At this grid level we only have physical boundaries. When integrating on grid level $l > 1$, not all boundary cells will abut the physical boundary (see Fig. 3.2). Those cells that do not abut the physical boundary are called internal boundary cells and for these cells additional boundary conditions have to be specified. Following Trompert and Verwer [63, 64] we prescribe, when necessary, Dirichlet boundary conditions for these cells which can be derived from the solutions

at time T and time $T + \Delta T$ at the underlying coarser grid level $l - 1$ by first applying spatial interpolation followed by temporal (linear) interpolation.

In case of advection it seems natural to have flux conditions at physical inflow boundaries. However, no information is available about these fluxes. We observed, for example, that assuming inflow of "clean" air may lead to unnecessary grid refinement in case the concentrations in grid cells next to boundary cells are higher than the computed concentrations in the boundary cells. For this reason, values in cells that abut the physical boundary are computed by extrapolation in case of inflow. We use constant extrapolation to prevent new extrema in the solution due to the extrapolation.

3.3.7 Mass conservation

We have seen that the grid refinement technique will disturb the mass conservation to a very limited extent. In general, this happens when the mass on the grid changes during the integration step due to in- and outflow. The change in mass is equal to the integral of the fluxes over the boundary of the computational grid.

To illustrate how mass conservation may be slightly disturbed, consider an integration step on a domain Ω using a coarse grid. On the subdomain $\Omega' \subset \Omega$ the coarse grid is refined. The exact change in mass ΔM on the subdomain is given by the integral of the fluxes over the boundary $\partial\Omega'$. On both the coarse and the fine grid this integral is approximated numerically in different ways, so in general, the approximations of ΔM will not be the same for both grids. Since the mass of the fine grid is injected into the coarse grid solution, the mass on the coarse grid will change with $\Delta M_F - \Delta M_C$ where ΔM_F and ΔM_C denote the change in mass on $\partial\Omega'$ using the fine grid and using the coarse grid, respectively.

Of course, it is possible to enforce exact mass conservation by imposing a suited flux condition at $\partial\Omega'$ for the integration on the fine grid. This condition may be derived from the fluxes over $\partial\Omega'$ during the coarse grid computation. However, this leads to a lot of extra overhead. In addition, negative values or undershoot may occur in advection computations. Further, the underlying coarse grid cells of the boundary cells of the fine grids have not been flagged by the space monitor, so strong gradients are not expected at fine grid boundaries. At this point the usefulness of the safety buffer, described in Section 3.3.2, becomes clear. Because we suppose the numerical solutions to be smooth at the fine grid boundaries no significant gain or loss of mass is expected due to the boundary treatment.

3.3.8 Time stepping

The model takes fixed overall time steps of half an hour (see Chapter 2). As operator splitting is applied, each subprocess is integrated separately. For each subprocess as many time steps can be taken as necessary to integrate from time level T to $T + \Delta T$. The number of time steps can also vary per grid level, as the integration on a fine grid is independent of integration on

the underlying coarse grid. For example, in case of advection, the time step on the base grid may be the overall time step, whereas on the finer grids two or more time steps are necessary in order to satisfy the Courant condition on these levels.

3.3.9 The datastructure

The datastructure is based upon the one described in [6] and is closely related to the one in [63]. Some modifications were necessary. The solution on each grid level (including the base grid) is stored row-wise in a one-dimensional array. Information about the structure of each grid level is also stored in a one-dimensional integer array. This array consists of several (sub)arrays that actually describe the grid:

- an array containing the number of rows in the grid and pointers to the start of each row,
- an array containing for each row the row number corresponding to its θ-coordinate,
- an array specifying for each grid cell its column number in a virtual rectangle, corresponding to its ϕ-coordinate,
- an array containing the number of physical and interior boundary cells and pointers to these cells,
- an array with for each grid cell a pointer to the underlying coarse grid cell it is part of,
- an array with for each grid cell a pointer to the cell directly above it,
- an array with for each grid cell a pointer to the cell directly below it,
- an array with for each grid cell a pointer to the lower left cell of the four cells on the next finer grid that form the present coarse grid cell.

In case cells, as indicated in the three last descriptions, do not exist, the corresponding pointers are set to zero. With the aid of the arrays listed above the implementation of numerical algorithms becomes only slightly more difficult in comparison to using just a uniform grid. Creating the fine grid structure is, in our opinion, not a very complicated task and, in our experience, only requires a few percent of the total computation time, including the construction of initial values and the injection procedure. The latter two processes can be implemented in a straightforward manner because of the pointers to underlying coarse grid cells and to finer cells on the next grid level. These pointers were not present in the datastructure of Trompert & Verwer [63, 64] and therefore they had to implement the interpolation of initial values and the injection procedure in a more complicated and time consuming way. For our application it is worthwhile to have these pointers, because we will have to perform injection relatively often due to the operator splitting approach.

For a more detailed description of the datastructure, see Appendix A.

3.4 Application to the four-layer model

In the previous section a description was given of the grid refinement algorithm for a two dimensional problem in space with only one solution component. The purpose of this section is to explain how we applied this technique within our smog prediction model with four layers in vertical direction and 15 species, described in Section 2.3. Recall that the model is not really a 3D model. The physical description is in terms of vertically averaged concentrations and therefore it makes no sense to refine in vertical direction. This explains why we restrict ourselves to a 2D refinement technique.

3.4.1 The grid

Each layer is numerically represented by a two-dimensional grid. Because of the several exchange processes between the layers (vertical diffusion, deposition, fumigation), it would be very inconvenient if in different layers different grids were created. An additional interpolation procedure would be necessary to interpolate the concentration function in the layer above and the layer below. Therefore, we work with the same grids in all layers. An additional advantage is that only one grid structure has to be created for all layers and components, thus reducing overhead and saving memory space.

3.4.2 When to refine?

As already mentioned, the numerical integration is performed in an operator splitting setting. In the odd integration steps we perform the subprocesses in the order: advection, diffusion, emission, deposition, fumigation and chemistry. In the even integration steps we perform them in reversed order, after an update of the model parameters. All processes are integrated on all grid levels. The remaining question is: should regridding take place in all processes or just in one of them? And, if we choose the latter, in which one? First of all, it is clear that avoiding regridding in each substep saves computation time. So it is worthwhile to look at each of the processes listed above and check whether they require regridding or not. Observing that all processes except advection and emission do not introduce (new) sharp gradients or move existing sharp gradients, we conclude that only advection and emission are candidates for regridding. As we can combine both by steering the grid refinement within the advection step, this step is the obvious choice. As mentioned earlier, refinement can be imposed not only according to criterion (3.2) but also to other, problem dependent criteria. If we also refine based on emission data, introduction of sharp gradients due to emission can be anticipated within the advection step. For another reason it also natural to have the refinement taking place in the advection step. In this step peaks in the solution are moved to other locations in the model area and the only possibility to follow this with grid refinement is by letting the advection determine the refinement.

3.4.3 The refinement criterion

The extension of criterion (3.1) and (3.2) to a four-layer model with 15 species is straightforward. First, we calculate the space monitor $spcmon(i,j)$ for each component i and each layer j according to (3.1). Then we calculate the total space monitor SPC

$$SPC = \max_{i,j} \left\{ \frac{w_{i,j} * spcmon(i,j)}{c_{i,j}^{max}} \right\}, \tag{3.10}$$

where the $w_{i,j}$ are weight factors corresponding to component i and layer j, with $0 \leq w_{i,j} \leq 1$; $c_{i,j}^{max}$ is the maximum value of component i in layer j on the grid level considered. Finally, the refinement criterion reads

$$SPC > tol, \tag{3.11}$$

where tol is a tolerance value to be defined by the user.

3.4.4 Computational efficiency

In case a new grid level is created in the advection step, a complete integration step including all subprocesses will be done on this new grid. In case of horizontal coupling between the grid cells the solution values in all grid cells on all grid levels are updated. In principle this is not necessary. In fact, only values in grid cells which are not further refined really need to be updated. All other values are obtained by injection from the next finer grid. However, for processes with horizontal coupling a complicated (and unphysical) procedure for the boundary conditions would be necessary since boundary conditions can no longer be derived from the just finished update on the next lower grid. Therefore, only in subprocesses with vertical coupling or no coupling at all between the grid cells, this (computational) advantage will be used. Its implementation is quite easy due to the pointer array pointing to cells on the next finer grid in case of further refinement, and to zero otherwise, see Section 3.3.9.

3.5 Numerical illustration

To show the effect of grid refinement, we consider scalar advection with the rotational wind field that will also be used in Section 4.6. After one rotation, the initial solution should be recovered. The wind components u and v are given by formula (4.56). One rotation is performed with a monotone advection scheme, i.e. a scheme that does not produce under- and overshoots in the solution. Hence, it makes sense to relate the accuracy to the maximum value on the grid. The related quantity is EMAX, defined by

$$\text{EMAX} = \frac{\max(C_i^0) - \max(C_i^1)}{\max(C_i^0) - \min(C_i^0)},$$

where the superscript indicates the number of rotations. The subscript i refers to cell i. Recall that due to the datastructure for the refined grids a single index is used to identify grid cells. The maximum value for the solution after one rotation is the maximum value over the solutions at all grids. Since grid refinement may disturb the mass balance on the base grid to some extent, we also consider the quantity ERR1

$$\text{ERR1} = \frac{\sum_i C_i^1 \cos\theta_i}{\sum_i C_i^0 \cos\theta_i} - 1,$$

which is only computed on the base grid.

As initial profile solution a Gaussian profile is considered, given by the function

$$c(\phi,\theta) = 1 + \exp\left\{-\frac{1}{2}((\phi-\phi_0)^2 + (\theta-\theta_0)^2)\right\}, \tag{3.12}$$

with ϕ and θ in degrees (shifted pole coordinates). The center of the Gaussian profile (ϕ_0, θ_0) is chosen such that its maximum coincides with a cell center on the base grid. The coordinates of this cell center are (6.325,-1.925). In Table 3.1 the results for this test are summarized. Since the spatial discretization in

	av. #cells used on level						
MAXLEV	1	2	3	4	use	EMAX	ERR1
1	2860	-	-	-	100	5.82e-1	9.66e-6
2	2860	519	-	-	30	2.61e-1	1.28e-4
3	2860	457	987	-	9	9.21e-2	8.36e-4
4	2860	663	1392	3218	4	4.52e-2	2.04e-3

Table 3.1: Results for the rotation test with the Gaussian profile

only first order at extrema in the solution, the value for EMAX is expected to decrease proportional to the mesh width, which is confirmed by the values in Table 3.1. The values of EMAX are almost equal for the same test performed on a uniform fine grid over the whole domain. The values for ERR1 then are of the order 10^{-5}. The column "use" indicates the percentage of cells actually used compared to the situation that the whole domain is covered by the finest grid. It shows that for this specific example grid refinement is very efficient. With respect to the mass balance, the table shows that this balance is disturbed indeed, but only to a very limited extent.

The same test is also performed for a block profile. The size of this block profile is chosen to be 2 × 2 grid cells on the base grid on exactly the same location as the profile used in the experiments in Section 4.6. The results for the block profile are listed in Table 3.2. This table shows again that the grid refinement procedure is very effective. For example, for MAXLEV=3,4 only a very small percentage of the total number of cells of a uniform fine grid is

MAXLEV	av. #cells used on level 1	2	3	4	use	EMAX	ERR1
1	2860	-	-	-	100	8.39e-1	7.51e-6
2	2860	345	-	-	28	6.03e-1	1.22e-4
3	2860	288	636	-	8	2.46e-1	2.44e-4
4	2860	273	547	1383	3	4.20e-2	8.87e-4

Table 3.2: Results for the rotation test with the block profile

used, 8 and 3%, respectively. The mass balance is disturbed, but again to a very limited extent.

In conclusion, the present illustration clearly shows the effectiveness of the grid refinement procedure. The numerical solution indeed becomes more accurate if grid refinement is applied, whereas the mass balance is only slightly disturbed.

Chapter 4

Finite-Volume Numerical Advection Schemes

4.1 Preliminaries

Horizontal advection is an important process in many atmospheric models. In this chapter we will focus on numerical advection algorithms, keeping in mind the application to our smog prediction model. Specific for this application is the regional scale of the model and the use of a grid refinement technique, see Chapter 3. The regional scale of the model makes it unnecessary to construct schemes that handle the singularity of the coordinate transformation at the poles. Therefore, the numerical algorithms in this chapter are formulated in usual Cartesian coordinates. They can easily be adopted for spherical coordinates. Only when necessary, it is indicated what should be done for application in spherical coordinates.

In [71] Williamson lists some desirable properties for advection schemes for use in global atmospheric models. From this list we consider relevant for our application:

- *Positivity:* Negative solution values may lead to instabilities when dealing with chemical equations. Therefore we require the scheme to be positive. It will turn out that for some numerical schemes the positivity property is closely related to the prevention of undershoot and often also overshoot. Although some undershoot (as long as the solutions remain nonnegative) or overshoot will not lead to chemical instabilities, it may generate errors which will act as perturbations for the initial values for the chemical integration. This obviously may disturb chemical equilibria and will make the solution of the chemical equations more difficult. However, we do not consider this to be very serious because the initial values for the chemistry will be out of equilibrium anyway due to the operator splitting and the update of parameters at the beginning of each integration interval.

- *Mass conservation:* Especially for long range transport it is important that mass is not (systematically) added to or deleted from the system, since then the dynamical behavior of the system can be disturbed to a large extent. For a smog model, predicting only a few days, this consideration is less relevant. Instead we could require that the mass balance

is not violated too much. However, for a scheme that is not strictly mass conserving there is no guarantee of a small conservation error, in particular when nearly discontinuous profiles have to be transported. The latter is the case in our model, due to emissions, so we prefer to stay on the safe side and restrict ourselves to mass conserving transport algorithms.

Since we already made the choice for finite-volume grid refinement, we confine ourselves to finite-volume schemes, for which the conservation property follows quite easily as we will see. Moreover, in this chapter we restrict ourselves to schemes derived along the method-of-lines (MoL). The algorithms described are the donor cell algorithm, a third-order upwind biased scheme with flux-limiting and flux corrected transport (FCT).

4.1.1 The advection equation

Cartesian coordinates

Horizontal dispersion of a pollutant in Cartesian coordinates is described by the following partial differential equation

$$\frac{\partial c}{\partial t} + \left[\frac{\partial (uc)}{\partial x} + \frac{\partial (vc)}{\partial y}\right] = 0, \tag{4.1}$$

where c denotes the concentration (field) of the pollutant and u and v wind velocities. The form (4.1) is sometimes called the *conservative form* of the advection equation. If the wind field is divergence-free, i.e.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{4.2}$$

equation (4.1) may be rewritten as

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} + v\frac{\partial c}{\partial y} = 0, \tag{4.3}$$

which is sometimes called the *advective form*. Some advection schemes start from the advective form, for example semi-Lagrangian schemes [41]. Note that in nature wind fields are divergence free, so one may use relation (4.2) even when starting from the conservative form. In practice we are only given values of u and v at certain points (e.g. cell centers), so we will assume a numerical equivalent of (4.2) to hold. The input wind fields of the model will not satisfy this numerical equivalent, however. Hence, a procedure is included to make the input fields numerically divergence free, see Section 4.5.

Spherical coordinates

In spherical coordinates, the advection equation takes the following form

$$\frac{\partial c}{\partial t} + \frac{1}{r\cos\theta}\left[\frac{\partial (uc)}{\partial \phi} + \frac{\partial (vc\cos\theta)}{\partial \theta}\right] = 0, \tag{4.4}$$

with ϕ and θ the longitude and latitude coordinate and r the radius of the earth. The expression for the divergence of the wind field now takes a slightly different form. The equivalent of (4.2) in spherical coordinates is

$$\frac{1}{r\cos\theta}\left[\frac{\partial u}{\partial \phi} + \frac{\partial(v\cos\theta)}{\partial\theta}\right] = 0. \tag{4.5}$$

The procedure used to make a wind field divergence free in Cartesian coordinates can be applied in spherical coordinates if we apply this procedure to $(\tilde{u},\tilde{v})=(u/r, v\cos\theta/r)$. The factor $\cos^{-1}\theta$ in (4.5) can be omitted since it only scales the actual divergences. In our application this term is not harmful because $\cos\theta$ never comes close to zero. If the wind field satisfies (4.5), the conservative form (4.4) may be rewritten as

$$\frac{\partial c}{\partial t} + \frac{u}{r\cos\theta}\frac{\partial c}{\partial \phi} + \frac{v}{r}\frac{\partial c}{\partial\theta} = 0. \tag{4.6}$$

Hence schemes that use the advective form need as velocities at the cell boundaries $(\tilde{u},\tilde{v})=(u/(r\cos\theta), v/r)$. The cosine term in $\tilde{u}$ represents that the physical mesh width in longitude direction decreases for an equidistant grid in spherical coordinates when approaching the poles.

4.1.2 Derivation of semi-discrete equations

The method-of-lines consists of two steps. In the first step, the spatial operator is discretized, resulting into a *semi-discrete system* because it is now discrete in space but still continuous in time. In the second step, the semi-discrete system is integrated in time with a suitable time integrator, resulting into a fully discrete system. In this section only the first step is discussed.

Cartesian coordinates

For the derivation of finite-volume schemes, we consider the conservative form of the advection equation. For ease of presentation we assume a rectangular spatial domain which we divide into rectangular cells $\Omega_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$, with cell centers (x_i, y_j).

In each cell Ω_{ij} we consider the average concentration C_{ij},

$$C_{ij}(t) = S^{-1}\int_{\Omega_{ij}} c(x,y,t)dxdy, \quad S = \Delta x\Delta y. \tag{4.7}$$

We will compute the time evolution of $C_{ij}(t)$ by means of a semi-discrete differential equation, which we derive from the conservative form of the advection equation by integration over Ω_{ij}

$$S^{-1}\int_{\Omega_{ij}}\frac{\partial c}{\partial t} = -S^{-1}\int_{\Omega_{ij}}\frac{\partial uc}{\partial x} - S^{-1}\int_{\Omega_{ij}}\frac{\partial vc}{\partial y}. \tag{4.8}$$

Interchanging integration and differentiation for the left-hand-side of (4.8) we arrive at

$$\frac{d}{dt}C_{ij}(t) = F_{i-\frac{1}{2},j} - F_{i+\frac{1}{2},j} + G_{i,j-\frac{1}{2}} - G_{i,j+\frac{1}{2}}, \tag{4.9}$$

where

$$F_{i\pm\frac{1}{2},j} = S^{-1}\int_{y_{j-1/2}}^{y_{j+1/2}} (uc)(x_{i\pm1/2}, y)dy \tag{4.10}$$

and

$$G_{i,j\pm\frac{1}{2}} = S^{-1}\int_{x_{i-1/2}}^{x_{i+1/2}} (vc)(x, y_{j\pm1/2})dx. \tag{4.11}$$

Usually, the integrals in (4.10) and (4.11) are approximated by the midpoint rule. We then obtain the approximations

$$\begin{aligned} F_{i\pm\frac{1}{2},j} &= (\Delta x)^{-1}(uc)(x_{i\pm1/2}, y_j), \\ G_{i,j\pm\frac{1}{2}} &= (\Delta y)^{-1}(vc)(x_i, y_{j\pm1/2}). \end{aligned} \tag{4.12}$$

The above approximations are second order accurate, provided that the values for uc and vc at the middle of the cell boundaries are approximated with at least second order accuracy. Higher order approximation is possible. For example, it can be verified that fourth order accuracy is obtained if we, after computing the fluxes in the above manner, simply put

$$F_{i\pm\frac{1}{2},j} \leftarrow \frac{F_{i\pm\frac{1}{2},j-1} + 22F_{i\pm\frac{1}{2},j} + F_{i\pm\frac{1}{2},j+1}}{24} \tag{4.13}$$

and for $G_{i,j\pm1/2}$ likewise, of course provided that uc and vc are approximated accurately enough. However, the schemes described in this chapter simply use (4.12). Note that for the evaluation of the Fs and Gs point values of the concentration function c are needed. They have to be approximated by the C_{ij} since these are the only quantities in the finite-volume context.

Spherical coordinates

In spherical coordinates, the surface S of a grid cell is also dependent on the latitude and is given by

$$S = r^2\Delta\phi\Delta\theta\cos\theta_j\delta,$$

with

$$\delta = \frac{\sin(\frac{1}{2}\Delta\theta)}{\frac{1}{2}\Delta\theta}.$$

We note that often the factor δ is omitted. This is reasonable, if $\Delta\theta$ is small enough, because then δ is very close to 1. See Section 3.3.4 for a similar discussion. On the base grid, $\delta = 0.999996...$ We therefore neglect this factor in all computations. Integrating the differential equation (4.4) and dividing by S results into the same semi-discrete system (4.9) as for Cartesian coordinates, but now with

$$F_{i\pm\frac{1}{2},j} = rS^{-1}\int_{\theta_{j-1/2}}^{\theta_{j+1/2}}(uc)(\phi_{i\pm1/2},\theta)d\theta \tag{4.14}$$

and

$$G_{i,j\pm\frac{1}{2}} = rS^{-1}\cos\theta_{j\pm1/2}\int_{\phi_{i-1/2}}^{\phi_{i+1/2}}(vc)(\phi,\theta_{j\pm1/2})d\phi. \tag{4.15}$$

Applying the midpoint rule for the integrals in (4.14) and (4.15) results into the flux expressions

$$F_{i\pm\frac{1}{2},j} = \frac{(uc)(\phi_{i\pm1/2},\theta_j)}{r\Delta\phi\cos\theta} \tag{4.16}$$

and

$$G_{i,j\pm\frac{1}{2}} = \frac{(vc)(\phi_i,\theta_{j\pm1/2})\cos\theta_{j\pm1/2}}{r\Delta\theta\cos\theta}. \tag{4.17}$$

4.2 The donor cell algorithm

The donor cell scheme is a combination of the first order upwind discretization for the advection operator and Forward Euler time integration. The scheme is described here for two reasons. The first reason is to illustrate the principle of positive discretizations and positivity of the fully discrete solution. The second reason is that this scheme will serves as a basic scheme for the FCT procedure described later on in this chapter.

Upwind and upwind biased schemes approximate uc and vc dependent from the direction of u and v. The donor cell algorithm approximates the fluxes according to

$$F_{i+\frac{1}{2},j} = \begin{cases} \dfrac{u_{i+\frac{1}{2}}}{\Delta x}C_{ij} & \text{if } u_{i+\frac{1}{2}} \geq 0, \\ \dfrac{u_{i+\frac{1}{2}}}{\Delta x}C_{i+1,j} & \text{if } u_{i+\frac{1}{2}} < 0. \end{cases} \tag{4.18}$$

The velocities $u_{i+\frac{1}{2}}$ at the cell boundaries are defined by $\frac{1}{2}(u_{i,j}+u_{i+1,j})$. The $G_{i,j+\frac{1}{2}}$ are defined in a similar fashion. We now show that the discretization (4.18) is positive, i.e.

$$\frac{d}{dt}C_{ij} \geq 0 \quad \text{if } C_{ij} = 0 \tag{4.19}$$

and all other values of C are nonnegative. From (4.18) we see that all negative contributions to dC_{ij}/dt are due to outflow from cell Ω_{ij}. On the outflow boundaries, the concentrations are approximated by C_{ij} itself, so the total outflow is zero. Only inflow in Ω_{ij} takes place, which is positive per definition.

The above condition is a necessary but not a sufficient condition for positivity in case of exact time integration. Since for application of Runge-Kutta methods to discretizations satisfying (4.19), conditions on the time step size

can be given for which the fully discrete solutions remain nonnegative, (4.19) is considered to be a sufficient condition for the space discretization throughout this chapter.

Positivity implies for the present discretization that the derivative dC_{ij}/dt will also be positive in case of a local minimum and negative in case of a local maximum. To show this, consider the transformation $W = \alpha C + \beta$. Since the discretization (4.18) is linear, we get

$$\frac{d}{dt}W_{ij} = \alpha\frac{d}{dt}C_{ij} - \beta D_{ij}, \tag{4.20}$$

where D_{ij} is a numerical approximation of the divergence of the wind field, given by

$$D_{ij} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} + \frac{v_{i,j+\frac{1}{2}} - u_{i,j-\frac{1}{2}}}{\Delta y}. \tag{4.21}$$

We will assume that D_{ij} is always zero. In practice, wind fields are made divergence free in such a way that this is true, see Section 4.5. With $D_{ij} = 0$, relation (4.20) states that positivity is equivalent with absence of undershoot, since C_{ij} is zero corresponds with a local minimum for W_{ij} ($\alpha > 0, \beta \neq 0$). It also shows that absence of undershoot is equivalent with the absence of overshoot, because for negative α, the time derivative of W_{ij} is zero in case of a local minimum in C_{ij}, i.e. a local maximum in W_{ij}. In conclusion, positivity implies the absence of under- and overshoot if the semi-discrete system is integrated in time exactly.

Unfortunately, the semi-discrete system cannot be integrated in time exactly. A numerical method has to be used to perform the time-integration. Positivity (in the wider sense of absence of under- and overshoot) of the discretization then turns out to be a necessary but not sufficient condition. In general the maximum step size will be restricted depending on the discretization and the specific time integration method.

This can be made clear by a simple calculation. Assume that u and v are constant in space and positive and suppose we integrate in time with the Forward Euler method. We then get the following scheme

$$C_{ij}^{n+1} = C^n + \nu_x(C_{i-1,j}^n - C_{i,j}^n) + \nu_y(C_{i,j-1}^n - C_{i,j}^n), \tag{4.22}$$

where the one-dimensional Courant numbers ν_x and ν_y are defined as

$$\nu_x = \frac{\tau u}{\Delta x}, \quad \nu_y = \frac{\tau v}{\Delta y}$$

with τ the time step size. The scheme (4.22) can be rewritten as

$$C_{ij}^{n+1} = (1 - \nu_x - \nu_y)C_{ij}^n + \nu_x C_{i-1,j}^n + \nu_y C_{i,j-1}^n,$$

which is a linear combination of the values of C with positive weights, so the scheme is free of under- and overshoot, provided that $1 - \nu_x - \nu_y \geq 0$. This is true if the two-dimensional Courant number ν satisfies

$$\nu = \nu_x + \nu_y \leq 1. \tag{4.23}$$

The result is in accordance with the result we found in [29], Section 3 with $\delta = 0$ and in Section 4.3.3 of this thesis. There we will also derive bounds on the Courant numbers for some Runge-Kutta methods for a class of discretizations of the advection operator with spatially varying velocity fields satisfying D_{ij}=0. In [29] only results have been derived for constant velocities. For the present discretization, however, it is not difficult to derive conditions for positivity in case of non-constant velocities. Observe that, due to D_{ij}=0, we have at least one and at most three outflow boundaries. The same holds of course for inflow boundaries. We derive conditions for positivity by simply examining all possibilities.

Suppose we have one outflow boundary. The three inflow boundaries give a positive contribution to C_{ij}^{n+1}, so our only concern is the outflow, given by νC_{ij}^n where ν is the 1D Courant number at this boundary. Since the total outflow must be restricted by C_{ij}^n it follows that the Courant condition for this case is $\nu_x, \nu_y \leq 1$.

Now suppose we have three outflow boundaries. The scheme can then be written as

$$C_{ij}^{n+1} = C_{ij}^n(1 - \tau D_{ij} - \nu) + \nu C_{kl}^n,$$

where ν represents the 1D Courant number at the only inflow boundary and the indices kl are different from ij. Since we assume D_{ij} to be zero, we again find $\nu_x, \nu_y \leq 1$ as condition for positivity.

The only case left is the situation of two inflow and two outflow boundaries. The scheme then reads

$$C_{ij}^{n+1} = C_{ij}^n(1 - \nu_1 - \nu_2) + \nu_1 C_{kl_1}^n + \nu_2 C_{kl_2}^n,$$

and it follows that all 1D Courant numbers must be smaller than $\frac{1}{2}$. Closer inspection reveals that this condition may be relaxed somewhat. If we assume that in both directions there is one outflow boundary, then a sufficient Courant number is $\nu_x + \nu_y \leq 1$, where both Courant numbers are taken at the outflow boundaries. If the two outflow boundaries are both in x-direction, the sum of both Courant numbers at the corresponding boundaries should be bounded by one. Since this implies a change of sign in u within cell Ω_{ij}, we may suppose that the velocities at the cell boundaries are so small that this condition is always met. The same holds if both outflow boundaries are in y-direction. Therefore we use as result for this case the condition $\nu_x + \nu_y \leq 1$.

Summarizing, we arrive at the following Courant condition for the first order upwind discretization with Forward Euler time integration

$$\nu_x + \nu_y \leq 1.$$

The first order upwind discretization combined with Forward Euler time integration is often called the *donor cell algorithm*. This algorithm (as well as the first-order upwind discretization together with high-order time integration) is not considered for application in the smog model, because it has

low accuracy and introduces an unacceptable amount of numerical diffusion. Yet it is relevant, because it is free of under- and overshoot. Therefore, it serves as a basis for the development of higher order flux-limited schemes.

4.3 The κ-discretizations

Since the first-order upwind discretization leads to a very diffusive scheme, we need a higher order approximation of the fluxes to obtain the desired accuracy. In this section we consider the family of κ-discretizations, introduced by Van Leer for application to the nonlinear Euler equations (see [36] and the references therein). The schemes are built from their one-space-dimensional forms. Therefore, for most of the discussion it suffices to consider the constant coefficient 1D problem. In the following subsection we will therefore describe the 1D case including the limiting procedure. Next the 2D discretization will be described and finally attention is paid to time integration aspects of the semi-discrete system. Most of what follows in this section comes from [28, 29].

4.3.1 1D formulation

We consider the 1D advection equation

$$c_t + (uc)_x = 0, \tag{4.24}$$

where for the moment we assume u to be constant and positive. The fluxes are defined as

$$F_{i+\frac{1}{2}} = \frac{u}{\Delta x} c_{i+\frac{1}{2}}, \tag{4.25}$$

where $c_{i+\frac{1}{2}}$ denotes the concentrations at the cell boundaries. They are approximated by

$$c_{i+\frac{1}{2}} = C_i + \frac{1-\kappa}{4}(C_i - C_{i-1}) + \frac{1+\kappa}{4}(C_{i+1} - C_i). \tag{4.26}$$

The values $\kappa = 1, -1$ and $\frac{1}{3}$ correspond with the second-order central, the second-order upwind and the third-order upwind biased discretization, respectively. With this discretization we would get the following semi-discrete system

$$\frac{d}{dt}C_i = u\left[\frac{\kappa-1}{4}C_{i-2} - \frac{3\kappa-5}{4}C_{i-1} + \frac{3(\kappa-1)}{4}C_i - \frac{\kappa+1}{4}C_{i+1}\right]. \tag{4.27}$$

It can easily be verified that no value for κ exists for which all weights in the stencil (4.27) are positive. Hence in this form the discretization is not positive and will give rise to under- and overshoots. To get a positive discretization, a limiting procedure must be developed. The first step in the procedure is to rewrite (4.26) as

$$c_{j+\frac{1}{2}} = C_j + \Phi(r_{j+\frac{1}{2}})(C_j - C_{j-1}), \tag{4.28}$$

where Φ is the limiter function. Its argument $r_{j+\frac{1}{2}}$ denotes the upwind ratio of consecutive solution gradients, defined as

$$r_{j+\frac{1}{2}} \equiv \frac{C_{j+1} - C_j}{C_j - C_{j-1}}. \tag{4.29}$$

If, for example, $\Phi(r)$ is taken equal to $K(r) = \frac{1}{6} + \frac{1}{3}r$, the unlimited $\kappa = \frac{1}{3}$-discretization is recovered. However, $\Phi(r)$ is supposed to work as an intelligent switch that defines the high order scheme as often as possible and only "limits" the high order flux if necessary to keep the discretization positive. Note that for $\Phi(r) = 0$ the first order upwind discretization is obtained. Since this discretization is positive we know that it is possible to obtain a positive discretization using the form (4.28). The term $\Phi(r_{j+\frac{1}{2}})(C_j - C_{j-1})$ can be viewed as a high-order correction term.

We now derive sufficient conditions for positivity. Using (4.28) the scheme reads

$$\frac{d}{dt}C_i = \frac{u}{\Delta x}\left[(1 + \Phi_{i+\frac{1}{2}}) - \frac{\Phi_{i-\frac{1}{2}}}{r_{i-\frac{1}{2}}}\right](C_{i-1} - C_i). \tag{4.30}$$

For positivity we require the bracketed term in (4.30) to be positive. In case C_i is a local minimum, the time derivative then becomes negative and in case C_i is a local maximum, the time derivative becomes positive. So here already we see the equivalence between positivity and the absence of over- and undershoot. We a priori assume $\Phi(r) \geq 0$ and $\Phi(r) = 0$ for $r \leq 0$. If $r_{i-\frac{1}{2}} \leq 0$ the discretization is positive because then $\Phi_{i-\frac{1}{2}} = 0$. For $r_{i-\frac{1}{2}} > 0$ the limiter values $\Phi_{i-\frac{1}{2}}$ and $\Phi_{i+\frac{1}{2}}$ should satisfy

$$\frac{\Phi_{i-\frac{1}{2}}}{r_{i-\frac{1}{2}}} - \Phi_{i+\frac{1}{2}} \leq 1. \tag{4.31}$$

This leads to the restriction

$$\Phi(r) \leq r.$$

Further, we require $\Phi(r)$ to be smaller than a constant $\mu > 0$. At this stage, μ is still arbitrary, but later on we will support the choice $\mu = 1$ because of the time integration. Summarizing, we impose the following constraint on Φ

$$0 \leq \Phi(r) \leq \min(r, \mu). \tag{4.32}$$

For the choice $\mu = 1$ the region (4.32) defines the total variation diminishing (TVD) region given in Fig. 1a of Sweby [61]. Note that the condition $\Phi(r) \leq r$ implies positivity of the approximations $c_{j+\frac{1}{2}}$ in (4.28) thus ensuring that the flux $F_{j+\frac{1}{2}}$ is of the same sign as $u_{j+\frac{1}{2}}$.

The limiter function $\Phi(r)$ should yield the unlimited, higher-order κ-discretization as often as possible, and therefore we define, following Koren [34],

$$\Phi(r) = \max(0, \min(K(r), r, \mu)), \tag{4.33}$$

where

$$K(r) = \frac{1-\kappa}{4} + \frac{1+\kappa}{4}$$

represents the original κ-discretization. In [28] we showed experimentally that the choices $\kappa = -1$ and $\kappa = 1$ lead to too diffusive results. As in [29, 28] we therefore only consider the choice $\kappa = \frac{1}{3}$.

The above analysis goes in exactly the same way for constant $u \leq 0$, leading to a similar result for the limiter function Φ. For completeness we give the flux expression

$$F_{i+\frac{1}{2}} = \frac{u}{\Delta x}(C_{i+1} + \Phi(\frac{1}{r_{i+\frac{3}{2}}})(C_{i+1} - C_{i+2})). \tag{4.34}$$

The definition of the limiter function then remains unchanged.

4.3.2 2D formulation

The 1D schemes are easily extended to the multi-dimensional case. The fluxes in each coordinate direction are approximated according to their 1D definition. In 2D this leads to the semi-discrete system (4.9) that is positive. For constant u and v this follows directly from the 1D definition of the fluxes. In 2D, however, we have to deal with spatially varying velocities, including the possibility of a change of sign in u or v within one grid cell. In each direction there are now four possible situations within a grid cell. Therefore, we consider again the 1D problem but now with non-constant $u = u(x)$ and examine the four situations on positivity.

Suppose C_i is zero and both the left and right cell boundary of Ω_i are outflow boundaries. Then both fluxes will be zero, either because the solution ratios are negative or because the solution in neighboring cells is zero as well. If both boundaries are inflow boundaries, the incoming fluxes are nonnegative by definition and thus $dC_{ij}/dt \geq 0$. Remains the case that both velocities have the same sign. Suppose they are positive. Because $\Phi_{i+\frac{1}{2}} = 0$, we then obtain

$$\frac{d}{dt}C_i = \frac{u_{i-\frac{1}{2}}}{\Delta x}\left[1 - \frac{\Phi_{i-\frac{1}{2}}}{r_{i-\frac{1}{2}}}\right] C_{i-1}.$$

We see that the condition $0 \leq \Phi(r) \leq r$ for $r > 0$ and $\Phi(r) = 0$ for $r \leq 0$ is sufficient for positivity. If both velocities are negative, the same result is obtained. The conclusion is that the 2D discretization for spatially varying velocities is positive, since the individual 1D contributions to the time derivative are positive.

To prove the equivalence between positivity and the absence of under- and overshoot, we apply the transformation (4.20) to the semi-discrete system. The solution ratios are invariant under the transformation. We therefore get, similar as for the upwind discretization,

$$\frac{d}{dt}W_{ij} = \alpha\frac{d}{dt}C_{ij} - \beta D_{ij}. \tag{4.35}$$

Since we assumed D_{ij} to be zero, the above expression implies the equivalence.

4.3.3 Time integration aspects

Preliminaries

In this section we shall discuss which explicit Runge-Kutta (RK) method can be used efficiently for the semi-discrete system (4.9) with the fluxes defined in Section 4.3.1.

The main criteria for the selection of a time integration method are *accuracy* and *positivity*: for reasonable Courant numbers the temporal error should not influence the total error too much and the solutions should remain nonnegative. Note that positivity together with mass conservation implies stability of the time integration, so stability need not explicitly be discussed. Unfortunately, numerical time integration may lead to negative solution values. Therefore conditions will be given under which the time integration together with the spatial discretization remains nonnegative. First we introduce some notation. The semi-discrete system is written as

$$\frac{d}{dt}C(t) = g(t, C(t))$$

where C (without subscripts) denotes the vector of concentrations on the grid. The function g is vector valued. Consecutive approximations $C^n \approx C(t_n)$ at time levels $t_n = t_0 + n\tau$, $n = 1, 2, \cdots$ are found by computing in each step internal vectors Y_i and their function values $G_i = g(t_n + \tau\,\eta_i\,, Y_i)$ according to

$$Y_i = C^n + \tau \sum_{j=1}^{i-1} a_{ij}\, G_j\,, \quad i = 1, 2, ..., s, \tag{4.36}$$

followed by

$$C^{n+1} = C^n + \tau \sum_{i=1}^{s} b_i\, G_i\,. \tag{4.37}$$

The method is thus determined by the real coefficients a_{ij}, b_i, η_i and the number of stages s. It can be compactly represented by the array

$$\begin{array}{c|c} \eta & A \\ \hline & b^T \end{array}$$

with lower triangular matrix $A = (a_{ij})$ and with $b = (b_i)$, $\eta = (\eta_i)$. In this section, we consider the following methods, represented below in Table 4.1 by their arrays. All methods in Table 4.1 have order p=s. The two 2-stage

0		
1/2	1/2	
	0	1

RK2a

0		
1	1	
	1/2	1/2

RK2b

0			
1/3	1/3		
2/3	0	2/3	
	1/4	0	3/4

RK3a

0			
1	1		
1/2	1/4	1/4	
	1/6	1/6	2/3

RK3b

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

RK4

Table 4.1: Arrays of the Runge-Kutta methods considered in this section.

methods are identical for linear problems. The same holds for the two 3-stage methods. Differences in the results are therefore caused by nonlinear phenomena. Note that the semi-discrete system obtained with limiting is highly nonlinear.

Experimental stability bounds

We find experimentally that for the *unlimited fluxes*, for which the semi-discrete system is linear, we have stability in 1D for Courant numbers

$$\nu \leq 0.87 \text{ for RK2a,b}, \quad \nu \leq 1.62 \text{ for RK3a,b}, \quad \nu \leq 1.74 \text{ for RK4}.$$

For the *limited fluxes* the stability bounds are found to be approximately

$$\nu \leq 1 \text{ for RK2a,b}, \quad \nu \leq 1.25 \text{ for RK3a,b}, \quad \nu \leq 1.4 \text{ for RK4}.$$

The values for the limited scheme are only approximately correct since the limited schemes show no very clear-cut transition from small errors to overflow. For the limited fluxes the experimental stability bounds must be greater than or at least equal to the bounds for positivity, since positivity together with mass conservation (which can be shown to be the case for the present discretizations and Runge-Kutta time integration methods considered here,) implies boundedness of the solution.

Theoretical bounds for positivity

We will now discuss some linear and nonlinear theoretical results on positivity. First we consider constant velocities and then generalize to spatially varying velocities. These results will be compared with experimental results.

For constant velocities, it suffices to consider the 1D problem with $u > 0$. The result in 2D then easily follows. In 1D the semi-discrete system can be written as

$$\frac{d}{dt}C_i = \gamma_i(C)(C_{i-1} - C_i) \tag{4.38}$$

with

$$\gamma_i(C) = \frac{u}{\Delta x}(1 + \Phi_{i+\frac{1}{2}} - \frac{\Phi_{i-\frac{1}{2}}}{r_{i-\frac{1}{2}}}). \tag{4.39}$$

It is easily verified that the condition (4.33) implies

$$0 \leq \gamma_i(C) \leq \frac{u}{\Delta x}(1 + \mu). \tag{4.40}$$

Applying the Forward Euler method (RK1) to the system (4.38) gives

$$C_i^{n+1} = C_i^n + \tau\gamma_i(C^n)(C_{i-1}^n - C_i^n)$$

and from (4.40) it follows directly that positivity is guaranteed under the condition

$$\nu \leq \nu_0 = \frac{1}{1+\mu}. \tag{4.41}$$

Here the parameter μ comes into play: even though the semi-discrete system is positive independent of the value of $\mu > 0$, the positivity condition for the fully discrete system depends on μ. The larger we choose μ, the more severe the restriction on the Courant number for RK1. This will also prove to hold for the other RK methods in Table 4.1. For these methods theoretical bounds that guarantee positivity can be obtained by following the approach of Shu and Osher [54, 55] on diminution of total variation (TVD). In this approach all stages of the Runge-Kutta method are written as convex combinations of Forward Euler type steps. Introducing

$$\alpha_{ij} \geq 0, \quad \sum_{j=1}^{i-1} \alpha_{ij} = 1, \quad \text{for } i = 2, ..., s+1, \tag{4.42}$$

the method can be written as

$$Y_i = \sum_{j=1}^{i-1} \alpha_{ij}(Y_j + \tau\frac{\beta_{ij}}{\alpha_{ij}}G_j), \quad i = 2, 3, ..., s+1, \tag{4.43}$$

with $Y_1 = C^n$, $C^{n+1} = Y_{s+1}$ and the coefficients

$$\beta_{ij} = a_{ij} - \sum_{k=j+1}^{i-1} \alpha_{ik} a_{kj}, \quad a_{s+1,j} := b_j. \tag{4.44}$$

If all $\beta_{ij} \geq 0$, it can be shown, just as for Euler's method, that we have positivity for Courant numbers

$$\nu \leq \nu_0 \min_{1 \leq j < i \leq s+1} \alpha_{ij}/\beta_{ij}.$$

Here ν_0 is the threshold value for Euler's method, and $\alpha_{ij}/\beta_{ij} = +\infty$ in case $\beta_{ij} = 0$. The result easily follows if we require all Forward Euler steps in (4.43) to be positive. The step sizes are equal to $\tau\beta_{ij}/\alpha_{ij}$ and each step is positive if $\nu \leq \nu_0 \alpha_{ij}/\beta_{ij}$. The minimum value of α_{ij}/β_{ij} thus determines the maximum allowable Courant number for positivity. Examining the RK-methods in Table 4.1 leads to the following Courant restrictions

$$\nu \leq \begin{cases} \dfrac{1}{1+\mu} & \text{for RK1, RK2b, RK3b,} \\ 0 & \text{for RK2a, RK3a, RK4.} \end{cases} \tag{4.45}$$

The above result on "nonlinear positivity" is based on worst-case assumptions for all stages. If we assume that $\gamma_i(C)$ in (4.38) remains almost the same over the stages, the situation will probably be described more accurately by a linear theory. Therefore, consider the system with "frozen coefficients"

$$\frac{d}{dt} C_i = \gamma_i (C_{i-1} - C_i), \qquad 0 \leq \gamma_i \leq \frac{u}{h}(1+\mu), \tag{4.46}$$

where $\gamma_i = \gamma_i(C(t_n))$ for $t_n \leq t \leq t_{n+1}$. On this system we can apply the linear theory of Bolley and Crouzeix [8]. From their Theorem 2 it can be deduced that we will have positivity for (4.46) under the condition $\nu \leq \nu_0/\xi$ where ν_0 is the threshold for Euler's method and ξ is the largest nonnegative number such that the stability function and all its derivatives are nonnegative on the interval $[-\xi, 0]$. In [35], Theorem 2.2, it was shown that $\xi = 1$ for any method having order $p = s$. Hence for all methods considered in this section we get the same condition for "linear positivity", namely

$$\nu \leq \frac{1}{1+\mu} \quad \text{for all methods in Table 4.1.} \tag{4.47}$$

The nonlinear results for RK2b, RK3b are thus optimal.

For 2D problems theoretical bounds for constant velocities can be obtained in a similar way. If $u, v > 0$, for example, the semi-discrete system can be written as

$$\frac{d}{dt} C_{ij} = \gamma_{ij}(C)(C_{i-1,j} - C_{ij}) + \delta_{ij}(C)(C_{i,j-1} - C_{ij}), \tag{4.48}$$

and the same conditions (4.45) and (4.47) as in 1D are obtained if we define

$$\nu = \tau \left(\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} \right). \tag{4.49}$$

Bounds for non-constant velocities

So far, we obtained theoretical bounds for constant velocities. We now generalize the result to non-constant velocities. Since the results on nonlinear positivity directly follow from positivity of RK1, we only need to examine RK1 on positivity. Again we only consider the 1D situation

$$C_i^{n+1} = C_i^n + \tau(F_{i-\frac{1}{2}} - F_{i+\frac{1}{2}}). \tag{4.50}$$

First we consider the situation that no change of sign in u occurs within Ω_{ij}. Suppose $u \geq 0$ in Ω_{ij}. Recall that $\tau F_{j+\frac{1}{2}}$ is then given by

$$\tau F_{i+\frac{1}{2}} = \nu \left(C_j^n + \Phi_{i+\frac{1}{2}}(C_{j-1}^n - C_j^n \right).$$

After some manipulations we obtain

$$C_i^{n+1} = (1 - \tau D_i^x)C_i^n + \left[\nu_{i+\frac{1}{2}} \Phi_{i+\frac{1}{2}} + \nu_{i-\frac{1}{2}}(1 - \frac{\Phi_{i-\frac{1}{2}}}{r_{i-\frac{1}{2}}}) \right] (C_{i-1} - C_i), \tag{4.51}$$

where

$$D_i^x = \frac{u_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}}{\Delta x}.$$

The term D_i^x is just a discretization of the derivative of u with respect to x in the cell center. If we assume that we get the derivative of v with respect to y from the fluxes in y direction, the sum of both is equal to D_{ij}, which we assumed to be zero. Therefore we act as if D_i^x is not present and concentrate on the bracketed term in (4.51). For positive weights of C_{i-1} and C_i we must have that this term is nonnegative and bounded by 1. Nonnegativity follows directly from $\Phi(r) \geq 0$ and $0 \leq \Phi(r)/r \leq 1$. Since also $\Phi(r) \leq \mu$ the bracketed term is bounded by $\nu_{i-\frac{1}{2}} + \mu\nu_{i+\frac{1}{2}}$. Requiring this bound to be at most one, we arrive at

$$\nu_{i\pm\frac{1}{2}} \leq \frac{1}{1+\mu}.$$

For negative velocities a similar result is obtained leading to the same condition on ν. Now we consider the case of outflow over the two cell boundaries. It turns out that C_i^{n+1} is an average of C_i^n, C_{i-1}^n and C_{i+1}^n. The coefficients for the latter two are positive. The coefficient for C_i^n is given by $1 - \nu_{i-\frac{1}{2}}\Phi_{i-\frac{1}{2}} - \nu_{i+\frac{1}{2}}\Phi_{i+\frac{1}{2}}$ which is positive if

$$\nu_{i\pm\frac{1}{2}} \leq \frac{1}{2\mu}.$$

This result seems somewhat strange, because the upper bound goes to infinity as μ approaches zero. Recall that for μ=0 we recover the first order upwind discretization. However, the result is partly due to explicit use of the fact that D_{ij}=0. It is therefore impossible that all four boundaries are outflow

boundaries and the Courant restriction in this case will come from the fluxes in y-direction.

The last situation we have to deal with is inflow over both cell boundaries. Since both fluxes give a positive contribution to the time derivative, we would expect no restriction on the Courant number from this case. This is not true, because we explicitly subtract τD_i^x when considering (4.50). The expression then becomes

$$\begin{aligned} C_i^{n+1} &= C_i^n + \nu_{i-\frac{1}{2}} \left[\frac{\Phi_{i-\frac{1}{2}}(r_{i-\frac{1}{2}})}{r_{i-\frac{1}{2}}} - 1 \right] (C_i^n - C_{i-1}^n) \\ &+ \nu_{i+\frac{1}{2}} \left[\frac{\Phi_{i-\frac{1}{2}}(r_{i+\frac{3}{2}}^{-1})}{r_{i+\frac{3}{2}}^{-1}} - 1 \right] (C_i^n - C_{i-1}^n). \end{aligned}$$

Since $0 \leq \Phi(r)/r \leq 1$, the weights for C_{i-1}^n and C_{i+1}^n are nonnegative. The weight for C_i^n is at least equal to $1 - \nu_{i-\frac{1}{2}} - \nu_{i+\frac{1}{2}}$ so the resulting restriction is

$$\nu_{i\pm\frac{1}{2}} \leq \frac{1}{2}.$$

It might be that this last result is somewhat unrealistic. This is however not important if we take $\mu \geq 1$ since then the other restrictions are more severe than the present one. All conditions are equivalent if we take $\mu = 1$.

Experimental bounds on positivity

Experimental results in 1D and 2D in [29, 28] give the experimental bounds for positivity as summarized in Table 4.2. The experiments confirm the

	RK2a	RK2b	RK3a	RK3b	RK4
1D	1	1	0.79	0.79	1.37
2D	0.66	0.67	0.86	0.78	<0.1

Table 4.2: Experimental ν-values for positivity with μ=1.

theoretical results in the sense that all experimental bounds are larger than the corresponding theoretical ones. It is interesting to see that the nonlinear theory is too pessimistic for RK2a and RK3a since this theory predicted zero bounds for these methods. The same holds for RK4, but much too our surprise this method failed to be positive in 2D, although the minima were small in absolute value.

4.3.4 Discussion of the κ-limiter

The limiting procedure as applied in the κ-schemes is, in our opinion, not optimal. In the first place because it is a one-dimensional limiter. All coordinate directions are discretized independently which may cause 'more limiting'

than really necessary. In the second place, even in one space dimension the limiter starts from worst-case assumptions. To make this point clear, consider the 1D discretization (4.30) with $u > 0$. Still assuming $\Phi(r) \geq 0$, the bracketed term in (4.30) is positive if

$$\Phi_{i-\frac{1}{2}} \leq (1 + \Phi_{i+\frac{1}{2}}) r_{i-\frac{1}{2}}. \tag{4.52}$$

Hence $\Phi_{i-\frac{1}{2}}$ and $\Phi_{i+\frac{1}{2}}$ are coupled. In the derivation of the limiter, however, we took the worst-case for $\Phi_{i+\frac{1}{2}}$ (which decouples both Φs) to arrive at $\Phi(r) \leq r$. This shows why the limiter is not optimal. The values of Φ can sometimes be chosen larger than the ones used by the present limiter. For the present 1D problem it is very simple to improve the limiting procedure by iteration on Φ using the relation (4.52). Numerical experiments show that this may lead to significant improvement of the solution. Note that this modification has no consequences for the positivity bounds since positivity results have been derived using $0 \leq \Phi(r) \leq \mu$ which is still valid. However, we do not propose a modification of the limiter along the lines described above because it becomes much more complex for 2D application. In that case we have variable velocities in the 1D discretizations with possibly different sign at two neighboring cell interfaces, so we then have to take too many possibilities into account. Moreover, the scheme would become more expensive due to the iteration on the Φs. Instead, we will consider a more general limiting procedure, *Flux Corrected Transport*, as described in the next section. This procedure is multi-dimensional and applicable to arbitrary flux expressions. The latter is an advantage over the κ-schemes, since we can easily include higher-order spatial discretizations.

4.4 Flux Corrected Transport

Flux corrected transport (FCT) is not so much an advection scheme as a limiting procedure. It has originally been developed by Boris and Book [10, 9, 11] and has been put in a generalized format by Zalesak [74]. The principle of FCT is very simple and it can be applied to arbitrary flux approximations. In the remainder of this section the FCT procedure will be explained in 1D. The procedure in multi-D then follows in a straightforward manner.

4.4.1 FCT in 1D

The basic assumption of the FCT procedure is the existence of low order fluxes $F^L_{i+\frac{1}{2}}$ such that the time-advanced solution $\tilde{C}_i^{n+1}$ with these fluxes

$$\tilde{C}_i^{n+1} = C_i^n + \tau(F^L_{i-\frac{1}{2}} - F^L_{i+\frac{1}{2}})$$

is positive, i.e. exhibits no under- and overshoot. From the previous subsections we know that such schemes exist. The fluxes F^L need not really be of low order, but usually low order fluxes are taken, e.g. defined by the donor cell algorithm, because their computation is cheap.

Suppose that we also have computed high-order flux approximations $F^H_{i+\frac{1}{2}}$ with the associated scheme

$$\tilde{C}_i^{n+1} = C_i^n + \tau(F^H_{i-\frac{1}{2}} - F^H_{i+\frac{1}{2}})$$

that is not positive. In order to make the scheme positive, FCT applies the following steps:

1. Define the so-called *anti-diffusive flux*

$$A_{i+\frac{1}{2}} = \tau(F^H_{i+\frac{1}{2}} - F^L_{i+\frac{1}{2}}).$$

2. Introduce the parameter $\gamma_{i+\frac{1}{2}}$ and assume that the flux to be applied is given by

$$\tau F_{i+\frac{1}{2}} = \tau F^L_{i+\frac{1}{2}} + \gamma_{i+\frac{1}{2}} A_{i+\frac{1}{2}}$$

with $0 \leq \gamma_{i+\frac{1}{2}} \leq 1$. In this way the flux will be a positive combination of a low-order approximation and a high-order approximation. The ideal situation is $\gamma_{i+\frac{1}{2}}$=1 because then the high order scheme is applied. So the procedure aims at choosing $\gamma_{i+\frac{1}{2}}$ as close as possible to 1.

3. Consider the total inflow in cell i caused by A given by the quantity P_i^+ and the maximum allowable inflow given by Q_i^+

$$\begin{aligned} P_i^+ &= \max(0, A_{i-\frac{1}{2}}) - \min(0, A_{i+\frac{1}{2}}), \\ Q_i^+ &= C_i^{max} - \tilde{C}_i. \end{aligned}$$

Here, C_i^{max} represents the maximum value for C_i^{n+1}. Because for physical reasons the solution is supposed to be free of overshoot, C_i^{max} may be defined as the maximum over some solution values close to cell i. For 1D problems one may for example define

$$C_i^{max} = \max(C_{i-1}^n, C_i^n, C_{i+1}^n, \tilde{C}_i^{n+1}).$$

The low order time-advanced solution $\tilde{C}_i^{n+1}$ is present because, depending on the definition of the low order fluxes, $\tilde{C}_i^{n+1}$ may be larger than the other three values. Including $\tilde{C}_i^{n+1}$ guarantees positive Q_i^+ and R_i^+.

The ratio R_i^+ between the quantities Q_i^+ and P_i^+ gives the maximum allowable inflow from A

$$R_i^+ = \min(1, \frac{Q_i^+}{P_i^+}).$$

4. Consider in the same way the total outflow from cell i due to A, resulting into the quantities C_i^{min}, P_i^-, Q_i^- and R_i^-.

5. Realizing that outflow from cell i is inflow in cell $i+1$ and vice versa, we can define the values of γ. It is easy to see that the choice

$$\gamma_{i+\frac{1}{2}} = \begin{cases} \min(R^+_{i+1}, R^-_i) & A_{i+\frac{1}{2}} \geq 0, \\ \min(R^+_i, R^-_{i+1}) & A_{i+\frac{1}{2}} < 0, \end{cases}$$

leads to positive approximations C_i^{n+1} if we finally put

$$C_i^{n+1} = \tilde{C}_i^{n+1} + \gamma_{i-\frac{1}{2}} A_{i-\frac{1}{2}} - \gamma_{i+\frac{1}{2}} A_{i+\frac{1}{2}}.$$

The above procedure is not optimal: it is possible that some values of γ could have been taken larger. This is because the total outflow from a cell is considered separately from the inflow. In order to achieve higher values of γ, the FCT procedure may be repeated a few times. This can be done by replacing the last step by

$$\tilde{C}_i^{n+1} = \tilde{C}_i^{n+1} + \gamma_{i-\frac{1}{2}} A_{i-\frac{1}{2}} - \gamma_{i+\frac{1}{2}} A_{i+\frac{1}{2}},$$

and by redefining A according to

$$A_{i+\frac{1}{2}} = (1 - \gamma_{i+\frac{1}{2}}) A_{i+\frac{1}{2}}.$$

After some iterations $\tilde{C}_i^{n+1}$ is accepted as the final solution and copied into C_i^{n+1}.

4.4.2 FCT in 2D

Extension of the FCT procedure as described above to 2D (and multi-D) is straightforward. For the computation of P^+_{ij} and P^-_{ij} (now all parameters involved get a double index) four values of A are considered instead of only two. For each flux, either in x-direction or in y-direction, a parameter γ is necessary, so we now have values $\gamma_{i+\frac{1}{2},j}$ and $\gamma_{i,j+\frac{1}{2}}$.

4.4.3 Application to MoL-schemes

In [74] a leapfrog scheme is used to integrate the semi-discrete system in time. For integration from t_n to $t_{n+1} = t_n + \tau$ this scheme requires the solution at t_n and $t_{n-1} = t_n - \tau$. Apart from the fact that this is only second-order accurate in time, the solution at t_{n-1} is not always available in practical applications. In the smog model, operator splitting is applied and therefore methods must be used that only need the initial values at the start of the integration interval, i.e. at time t_n. Since for advection on the base grid only one time step equal to the overall time step of the operator splitting is necessary, we concentrate on time integration with Runge-Kutta methods. Suppose we have a high order spatial discretization and we integrate with one of the RK methods from Section (4.3.3), we have to derive one flux expression from the final result. This turns out to be very easy. For ease of presentation, we consider once more the 1D case and we observe that for RK methods the final approximation is written as

$$C_i^{n+1} = C_i^n + \tau \sum_{k=1}^{s} b_k (F^k_{i-\frac{1}{2}} - F^k_{i+\frac{1}{2}})$$

where the upper index k of the fluxes F corresponds to the flux approximation used in the k-th stage. From this expression it follows that the flux over the cell boundaries are in fact approximated by

$$F^H_{i+\frac{1}{2}} = \sum_{k=1}^{s} b_k F^k_{i+\frac{1}{2}}$$

which is the desired flux approximation for the FCT procedure. In contrast to the application of RK methods for the limited κ-discretizations, all RK methods lead to positive schemes when FCT is applied. The procedure has only to be applied once per time step whereas the limiting procedure in the κ-scheme has to be done in each stage.

4.4.4 Application in spherical coordinates

The procedure is easily modified for application in spherical coordinates. We just compute the fluxes and the anti-diffusive fluxes as if the factor $\cos\theta$ in (4.4) is not present. The ratios R^+_{ij} and R^-_{ij} are then computed according to

$$R_i^{+/-} = \min(1, \frac{Q_i^{+/-} \cos\theta_j}{P_i^{+/-}}).$$

The desired result now follows immediately.

4.4.5 Discussion of FCT

The FCT algorithm is a very flexible one, because it can be applied to any flux expression. Especially for MoL-schemes this seems to be attractive, since limiting procedures for such schemes are usually dependent on the discretization. Another advantage of FCT over e.g. the κ-limiter is its multi-D nature. Only inflow and outflow are decoupled. The coupling may be restored by a simple iteration procedure. In Section 4.6 results of a numerical comparison is presented between schemes with flux-limiting and schemes using FCT.

4.5 Making wind fields divergence-free

4.5.1 Preliminaries

In this section we describe how in the model wind fields are made divergence-free. This is achieved by following the procedure proposed in [18]. We describe here the procedure in Cartesian coordinates, but - as pointed out in Section 4.1.1 - the same procedure can be applied for spherical coordinates without modification, provided that the model domain does not contain a pole and the cosine of the latitude does not become too small. Due to introduction of the shifted pole coordinates, the cosine of the latitude on our

domain has a smallest value of approximately 0.92 at the most Southern edge of the domain, i.e. at -23.1^o.

In nature wind fields are divergence free. Recall that divergence-freeness of a wind field (u, v) implies

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \tag{4.53}$$

If a wind field satisfies the above condition, the exact solution of the advection equation exhibits no under- and overshoots. For several numerical advection schemes it can be proved that the numerical solutions do not exhibit under- and/or overshoot, provided that a numerical equivalent of (4.53) holds. In this section we consider as numerical equivalent for (4.53) the divergence D_{ij} in grid cell Ω_{ij}

$$D_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}, \tag{4.54}$$

assuming that we are given values of (u, v) in the cell centers. For ease of presentation, the procedure is explained on a rectangular base grid. The procedure can be straightforwardly implemented on grid structures as created by the refinement algorithm from Chapter 3. In the description of the numerical advection schemes in this chapter, it has been shown if and when this numerical equivalent prevents under- and overshoots.

4.5.2 Necessity of the procedure

The wind fields in the model generally do not satisfy $D_{ij} = 0$. Even worse, this relation is often so heavily violated that the result of an advection step is very inaccurate. This is illustrated by Figure 4.1 where a solution plot is given after an integration step of half an hour using the wind field of 19 July 1989, 12:00 GMT, starting with a uniform concentration distribution. The

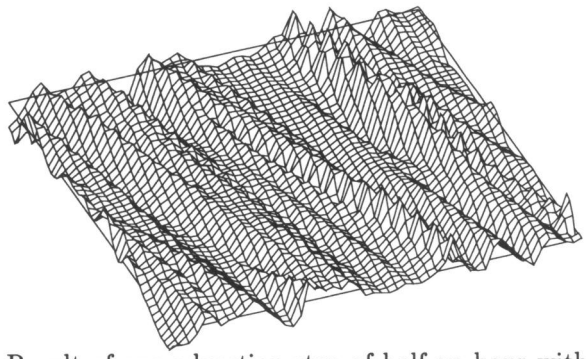


Figure 4.1: Result of one advection step of half an hour with the divergent input wind field for 19 July 12:00 GMT.

deviation of the uniform concentration is between -10% and 10% and it is

clear that the resulting solution has no physical meaning. This is not due to the advection scheme used, but due to the wind fields themselves. We cannot explain the strange wave pattern in Figure 4.1. Maybe this pattern is due to the fact that the ECMWF model is a spectral model. We suggest further investigation on deriving wind fields for the smog model, based on knowledge of the underlying ECMWF wind fields, because wind fields are one of the most important input data for the smog model. In Chapter 7 where model results will be compared with measurements, good wind data will prove to be of great importance.

The wind fields used in the model are derived from ECMWF wind fields by spatial bi-linear interpolation. The resolution of the ECMWF field is only $3^o \times 3^o$ in standard lat-lon coordinates, which is much coarser than the resolution used in our model. Apart from the fact that the original ECMWF fields do not satisfy (4.54) to be zero, the interpolation procedure may cause even a more severe violation of $D_i j = 0$.

4.5.3 The procedure of Endlich

The first part of the procedure of Endlich [18] consists of iterative application of the following two steps

1. Compute D_{ij} for $i = 1, \ldots, N$, $j = 1, \ldots, M$.

2. Compute for $i = 1, \ldots, N$, $j = 1, \ldots, M$ the "new" values of the wind field

$$\begin{aligned} u_{i+1,j} &= u_{i+1,j} - \tfrac{1}{4}\Delta x D_{ij} \\ u_{i-1,j} &= u_{i-1,j} + \tfrac{1}{4}\Delta x D_{ij} \\ v_{i,j+1} &= v_{i,j+1} - \tfrac{1}{4}\Delta y D_{ij} \\ v_{i,j-1} &= v_{i,j-1} + \tfrac{1}{4}\Delta y D_{ij} \end{aligned}$$

If step 1 and 2 are applied for a single cell only, the numerical divergence (4.54) would become zero. However, in general each value of u and v is modified more than once when looping over all grid cells, so iterative application of step 1 and 2 is applied to reduce the divergence. The iteration is stopped if the maximum value of D on the grid is smaller than some parameter ε. In the model we take ε in the range $[10^{-7}, 10^{-6}]$.

If the iterative part is terminated, all velocities (i.e. $\sqrt{u^2 + v^2}$) are scaled such that the average velocity of the original field is retained. This only affects the maximum of D_{ij} to a limited extent.

To see what happens if we apply the procedure to a given wind field, in Fig. 4.2 a wind field together with the resulting divergence free wind field is plotted. No structural changes are visible. It took 491 iterations with $\varepsilon = 10^{-7}$. The average difference between the individual wind velocities was about 12% and the angle between the wind vectors before and after application of the procedure was about 25.3^o on average. We consider these

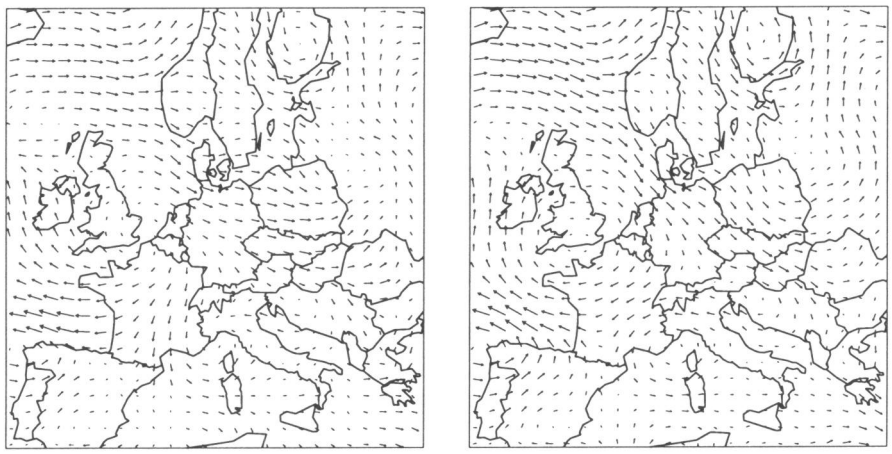


Figure 4.2: Wind field of 19 July 1989, 12:00 GMT. Left: wind field derived from ECMWF field by interpolation. Right: the same wind field made divergence-free.

numbers to be (too) large, but it seems that making wind fields divergence-free according to methods like the one described here, is the only choice. This stresses the necessity of further investigation.

4.5.4 Connection with advection schemes

In the description of the advection schemes we required a different approximation for the divergence to be zero (i.e. $\leq \varepsilon$) than the approximation (4.54) from this section. For advection schemes we used

$$D_{ij} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y}. \tag{4.55}$$

It therefore depends on the definition of the velocities at the cell boundaries whether (4.55) is small. In practice, linear interpolation is used to obtain values for the wind velocities at the cell boundaries

$$\begin{aligned} u_{i+\frac{1}{2},j} &= \frac{1}{2}(u_{i+1,j} + u_{i,j}), \\ v_{i,j+\frac{1}{2}} &= \frac{1}{2}(v_{i,j+1} + v_{i,j}). \end{aligned}$$

Substituting this choice into (4.55) we reobtain expression (4.54).

4.6 Numerical experiments

In this section a numerical comparison is given for the MoL scheme from this chapter with flux limiting and FCT to prevent under- and overshoot.

Also the direct scheme from [31, 30] is considered. The methods are tested in the same way as they would be applied in the smog model. Hence we present results for advection in spherical coordinates, on the model domain described in Chapter 2 and use the same base grid. Hence, we have a $N \times M$ grid, with $N = 52$ and $M = 55$. For this purpose, two model problems have been designed. For both problems, as initial solution block profiles are taken. Profile 1 consists of 2×2 grid cells, profile 2 of 4×4 grid cells and profile 4 of 8×8 grid cells with height 1, relative to a background value of 1. They are always located such that, with exact advection, the (possibly deformed) profiles stay within the domain. The fluxes at inflow boundaries are then simply given by the velocity normal to the cell boundary times the background concentration.

Problem I: Molenkamp test
In spherical coordinates the rotational wind field is given by (see e.g. [72])

$$\begin{aligned} u &= U(\cos\beta\cos\theta + \sin\beta\sin\theta\cos\phi), \\ v &= -U(\sin\beta\sin\phi), \end{aligned} \tag{4.56}$$

where β is the angle between the polar axis and the axis of the rotation. U is a scaling factor. For $U = 1$ the period of one rotation is $2\pi r$ seconds, r being the radius of the earth in meters. For testing purposes this wind field has the advantage over its Cartesian twin that u and v are not constant in ϕ- resp. θ-direction. In order to enable a rotation within the model domain, β in (4.56) is taken approximately 82^o. To get the center of the rotation in the middle of the domain, u and v are evaluated in $(\tilde{\phi}, \theta)$ with $\tilde{\phi} = \phi - 6.05^o$. The 2×2 block profile is obtained by assigning the value 2 to the concentration in the cells Ω_{ij}, $i = 25, .., 28$ and $j = 42, .., 45$. The larger block profiles have the same center as the 2×2 profile. The time step is taken such that 1 rotation takes 500 time steps.

Problem II: Wind field from database
Since we have to select one or more possible schemes for application in the model, we also perform some tests with wind fields from the database that is used by the model. Wind fields from the July 1989 and November/December 1989 episodes will be used (see Chapter 7). Instead of updating the wind fields each hour, we just take a wind field and keep it constant during the test. The following wind fields have been selected. The wind field selected is the 1000mbar field from 22-07-1989, 12:00 GMT. A vector plot of this wind field is given in Figure 4.2. For this problem we only considered the 8×8 block profile, which has been placed approximately in the middle of the domain. The value 2 is assigned to the concentrations in the cells Ω_{ij}, $i = 25, .., 28$ and $j = 24, .., 31$, and 1 to all other cells. The time step is taken half an hour, as in the smog model and 192 time steps are taken, so that the total integration interval is 4 days. Of course, in reality the wind field changes with time (in the model it is updated each hour and then kept constant for an hour), but for this test we kept the wind field constant during the two days. Then we reverse all velocities and integrate for another 2 days, so that at the end of the four days interval the initial profile should be restored.

The methods considered and their application to the test problems are listed below

1. The limited $\kappa = \frac{1}{3}$ discretization described in Section 4.3. For the time integration RK3b from Section 4.3.3 is taken.

2. The unlimited $\kappa = \frac{1}{3}$ discretization, also with RK3b time integration. FCT is applied to prevent under- and overshoots.

3. The spatial discretization is the fourth-order central discretization. Hence, the fluxes $F_{i+\frac{1}{2},j}$ are approximated by

$$F_{i+\frac{1}{2},j} = u_{i+\frac{1}{2},j} \frac{7(C_{i,j} + C_{i+1,j}) - C_{i-1,j} - C_{i+2,j}}{12\Delta\phi},$$

   and similarly for $G_{i,j+\frac{1}{2}}$. Again with RK3b time integration and FCT to prevent under- and overshoot.

4. The limited dimensional split scheme described in [31, 30]. The so-called Strang splitting [59, 40] is used. First a half time step is performed in ϕ-direction, then a full time step in θ-direction and finally again a half time step in ϕ-direction. This leads to second order accuracy in time, provided that the time steps in in ϕ- and θ-direction are also with at least second order accuracy in time. This is achieved by the application of modified velocities at the cell boundaries, see [31, 30].

5. The same scheme as in 4, but with unlimited fluxes. FCT is used to prevent under- and overshoot.

For the schemes that use FCT, the low order fluxes are defined by the donor cell algorithm. The number of FCT iterations is taken equal to one. We note that the split scheme can be applied in a more efficient way. The velocities can be modified such that performing a full time step in both coordinate direction already results into second-order accuracy in time. We do not apply the scheme in this way because it is a complex modification on refined grids and it involves extra storage of modified wind fields. Since an advection step in ϕ-direction is cheaper than an advection step in θ-direction, the present way of application of the split scheme is less than 1.5 times more expensive than in case of the more efficient way of application.

To measure errors and accuracy, the following quantities are considered, similar as in [29].

$$\begin{aligned}
\text{EMAX} &= \frac{\max(C_{ij}^n) - \max(C_{ij}^0)}{\max(C_{ij}^0) - \min(C_{ij}^0)}, \\
\text{EMIN} &= \frac{\min(C_{ij}^n) - \min(C_{ij}^0)}{\max(C_{ij}^0) - \min(C_{ij}^0)}, \\
\text{ERR0} &= \frac{\sqrt{\sum_{ij}(C_{ij}^n - C_{ij}^0)^2}}{N \times M(\max(C_{ij}^0) - \min(C_{ij}^0))}, \\
\text{ERR1} &= \frac{\sum_{ij} C_{ij}^n \cos\theta_j}{\sum_{ij} C_{ij}^0 \cos\theta_j} - 1.
\end{aligned}$$

The supersripts 0 and n refer to the initial and final solution. Hence, a negative value for EMAX means that the maximum value on the grid has decreased during the advection. If EMIN is negative, the scheme has produced negative values. ERR0 is the scaled L_2 error. The error in the total mass is measured by ERR1 (recall that the surface of a grid cell is proportional to $\cos\theta_j$). Since all methods considered should be mass conserving, ERR1 can only be different from zero due to in/outflow. The tests are chosen such that for the exact solution outflow and inflow are equal. Mass difference can therefore only be caused by numerical diffusion over the domain boundary. We assume small values of ERR1 ($\leq 10^{-12}$) to be caused by rounding errors and its value will be reprented by zero. To measure efficiency, CPU times for all experiments are specified. These timings only serve as an indication of the efficiency, because they are machine dependent. They have, however, been carried out on a SGI workstation. The codes have been compiled with the -O option.

4.6.1 Results for Problem I

In Table 4.3, the results for Problem I on the base grid are summarized. From the CPU times we conclude that FCT is a relatively expensive procedure, compared to flux limiting. This becomes clearest if we look at the results for method 4 and 5. The difference in CPU time is about 7 seconds, whereas the only difference is that method 4 uses the limited fluxes (and hence no FCT) and method 5 the unlimited fluxes and applies FCT. Hence, the flux computations in method 5 are somewhat cheaper than in method 4. The fact that method 5 takes more CPU time is thus caused by the FCT procedure including the additional first order upwind fluxes as basic scheme for the FCT. The same observation can be made for method 1 compared with method 2 and 3. The difference in CPU time is not as large as the difference between method 4 and 5 because the flux computations for method 1, 2 and 3 are more expensive. Instead of two flux evalutions in ϕ-direction and one in θ-direction, in these methods 3 flux evaluations in both coordinate directions are needed because of the RK3b time integration. Computing the unlimited fluxes for method 2 and 3 thus saves more CPU time compared to the limited

profile	method	EMAX	EMIN	ERR0	ERR1	CPU
1	1	-0.92	0	6.60e-4	-7.22e-12	14.0
	2	-0.89	0	3.43e-4	-8.21e-09	18.3
	3	-0.83	0	6.41e-4	-6.60e-08	16.3
	4	-0.89	-6.27e-4	6.47e-4	4.32e-04	9.0
	5	-0.88	0	6.40e-4	-4.60e-07	16.9
2	1	-0.69	0	1.09e-3	-6.93e-11	14.0
	2	-0.58	0	9.97e-4	-3.50e-09	18.3
	3	-0.37	0	8.28e-4	-2.10e-07	16.3
	4	-0.60	-6.27e-4	1.02e-3	4.30e-04	9.0
	5	-0.56	0	9.84e-4	-4.81e-07	16.9
3	1	-0.14	0	1.32e-3	-1.07e-8	14.0
	2	-0.01	0	1.26e-3	-1.33e-7	18.3
	3	-0.03	0	1.07e-3	-1.43e-6	16.3
	4	-0.02	-6.27e-4	1.26e-3	4.23e-4	9.0
	5	-0.005	0	1.25e-3	1.39e-7	16.9

Table 4.3: Results for problem I on the base grid

method	1	2	3	4	5
fluxes	13.0 (88%)	5.9 (32%)	4.4 (25%)	8.5 (92%)	4.4 (25%)
FCT	-	10.6 (57%)	10.6 (61%)	-	12.0 (70%)

Table 4.4: CPU times for the flux computations and the FCT procedure for the 5 schemes.

fluxes than for method 5. Since FCT is applied only once for method 2, 3 and 5, the increase in CPU time will therefore be smaller for method 2 and 3. This is also illustrated by Table 4.4 where results of timings for the 5 methods are summarized. For all methods the time needed for the flux computations have been measured as well as the time needed for the FCT procedure, including the computation of the first order upwind fluxes, is specified.

Applying FCT to the MoL scheme with unlimited 3rd order upwind fluxes (method 2) gives smaller values for EMAX and ERR0 than the MoL scheme with 3rd order limited fluxes (method 1). Only ERR1 is somewhat larger, but still acceptable. Using fourth order central flux approximations and FCT (method 3) instead of 3rd order, gives some improvement, only ERR1 becomes somewhat larger. Surprisingly, the CPU time for method 3 is lower than for method 2. The only difference between these methods is the flux computation. For the third-order discretizations, the direction of the wind needs to be checked. This is implemented using an 'IF'-construction, which is apparently responsible for the difference in CPU time.

Although the split scheme with limited fluxes (method 4) is the cheapest method per time step, it produces small undershoots. Thus, if strict positivity (i.e. prevention of under- and overshoots) is necessary, this method cannot

be used. Also ERR1 is a number of orders of magnitude larger than for the other methods, but still acceptable. Applying the unlimited fluxes with FCT (method 4) satisfies the positivity requirement, EMAX and ERR0 are comparable, but ERR1 has become much smaller. However, the CPU time almost doubles and method 5 is now even more expensive than method 1 and 3.

To see the effect of grid refinement, we simply doubled the number of grid cells in ϕ- and θ-direction. The surface of the block profiles have the same size as for the tests on the base grid, and are located at the same place within domain. Hence, the 2×2 profile on the base grid is identical to a 4×4 profile on the refined grid etc. The time step has been halved for all methods so that the local Courant number are the same as in the tests on the base grid. The results for the refined grid are summarized in Table 4.5.

profile	method	EMAX	EMIN	ERR0	ERR1	CPU
1	1	-0.72	0	3.76e-4	0	111.3
	2	-0.61	0	3.84e-4	8.07e-06	159.7
	3	-0.43	0	4.01e-4	8.08e-08	143.3
	4	-0.64	-1.56e-4	3.83e-4	2.94e-04	71.1
	5	-0.61	0	3.85e-4	8.07e-06	144.5
2	1	-0.19	0	9.09e-2	0	111.3
	2	-0.04	0	9.26e-2	3.21e-5	159.7
	3	-0.03	0	9.50e-2	3.21e-5	143.3
	4	-0.06	-1.56e-4	9.23e-2	2.93e-4	71.1
	5	-0.04	0	9.27e-2	3.21e-5	144.5
3	1	-7.47e-5	0	1.95e-1	0	111.3
	2	-1.22e-9	0	1.96e-1	1.26e-4	159.7
	3	-5.03e-5	0	1.99e-1	1.26e-4	143.3
	4	6.66e-4	-1.56e-4	1.96e-1	2.88e-4	71.1
	5	-6.10e-10	0	1.97e-1	1.26e-4	144.5

Table 4.5: Results for problem I with double resolution

Most of the conclusions for the the results on the base grid are still valid. For profile 2 and 3, the values for EMAX become neglectable for all methods. Also the values for ERR0 are of comparable size for all methods. The same holds for ERR1, except for method 1 which has a smaller value than the other methods. This could already be concluded from Table 4.3.

4.6.2 Results for Problem II

The results of the experiments for problem II are given by Table 4.6. The results confirm the findings in the previous subsection for Problem I. Note that the absolute values for EMIN for the methods is much larger. This is because the numerical divergence, which is about 10^{-13} for Problem I, now is in the order of 10^{-8}. For example, the undershoot of the donor cell algorithm (the basic scheme for method 2, 3 and 5) that can be made in

method	EMAX	EMIN	ERR0	ERR1	CPU
1	-0.27	-3.82e-5	1.57e-3	1.08e-05	5.3
2	-0.16	-6.03e-4	1.40e-3	9.55e-06	7.0
3	-0.06	-4.89e-4	1.15e-3	7.84e-06	6.2
4	-0.19	-1.50e-2	1.46e-3	-2.06e-04	3.5
5	-0.16	-6.03e-4	1.39e-3	-1.03e-05	6.4

Table 4.6: Results for problem II

one step, is the numerical divergence times the time step size multiplied by the concentration. The time step is 1800 seconds and the concentrations are $O(1)$, so the undershoot due to one time step can be of the order 10^{-5}. The number of time steps is equal to 192. The undershoot can therefore be in the order of 10^{-3}. Of course, the latter result is based on a worst case assumption, that will probably not occur in practice. The values in Table 4.6 are of order 10^{-4}, which is quite reasonable. Only the value of EMIN for method 4 is two orders of magnitude larger. Since this number is already 1.5% of the background concentration and no bounds for the under- and overshoots can be given for this method, we do not consider method 4 to be suited for application in the smog model.

4.6.3 Conclusions

From the experiments reported above, the following conclusions are drawn

1. The split scheme described in [31, 30], is the most efficient method of the methods considered. However, because it may produce substantial undershoots (although positivity of the solutions is guaranteed) this method is not suited for application in the smog model.

2. From the remaining methods, method 1 is the cheapest method, but not the most accurate one. Method 3 seems the most accurate one and is about 15% more expensive than method 1.

3. If grid refinement is applied, the differences in the results of the methods are expected to become smaller. Since FCT is a relatively expensive procedure, that is expected to become even more expensive if it is implemented on refined structures, we chose to implement method 1 in the smog model.

Chapter 5

Chemical Solution Methods

5.1 Introduction

Atmospheric models are computationally very expensive. Usually the computational work is dominated by the numerical treatment of the systems of ordinary differential equations (ODEs) describing the chemical transformations. In large scale models sometimes 80% or more of the total computation time is spent on solving these ODEs. One reason for this is that these systems are *stiff* due to the simultaneous presence of slow and very fast reacting species. Another reason is that, in every time step, the solution of the chemical equations is required at all grid cells. This explains the need for fast and efficient special purpose solvers. Fortunately, the accuracy level is quite modest, say at most 1%. In atmospheric models a higher accuracy is considered to be redundant because of various other uncertainties about physical parameters and input data. Therefore it suffices to solve the chemistry at a low accuracy only. In this chapter some special purpose solvers are described and examined for use in our smog prediction model. Also a comparison is provided between these special purpose solvers and the state-of-the-art code VODE from the numerical stiff ODE field. It will be shown that the special purpose solvers do outperform VODE for the present application. Hence, the emphasis is on special purpose solvers which satisfy the accuracy requirement of 1% and are very fast as well. Such methods exploit the special form of the chemical kinetics system

$$\dot{y}(t) = f(t,y) \equiv P(t,y(t)) - L(t,y(t))y(t), \tag{5.1}$$

with given initial vector $y(0)$. P is a k-vector specifying the production terms (possibly including source terms) and L a $k \times k$ diagonal matrix defining the loss rates (possibly including sink terms), k being the number of species in the system. By definition P and L are nonnegative for all t and nonnegative y. In the remainder of this chapter we will often use the autonomous form for the functions P, L, f etc. and omit time as argument.

In this chapter we will consider two special purpose solvers, including some possible variants. The first solver is TWOSTEP, developed by Verwer [65, 70]. The second solver is based on the Quasi Steady State Approximation (QSSA) approach. Both solvers will be compared to the state-of-the-art solver VODE [12, 16]. We exploited the sparsity of the Jacobian within VODE by replacing its linear algebra routines by appropriate sparse matrix routines.

5.1.1 Mass conservation, positivity and stability

In the description and discussion of the solvers attention will be paid to the (lack of) positivity and conservation of mass. As we will see, there is a relation between the latter two and stability.

Positivity of the solution components is natural from the chemical point of view and desirable for numerical reasons. Especially special purpose solvers use the nonnegativity of L and negative components of L may cause stability problems.

Especially for long range transport models it is of importance that mass is not systematically added to or deleted from the system. However, in our short term prediction model some gain or loss of mass due to numerical integration may be of less importance. In case the exact solution of system (5.1) is mass conserving in the sense that one or more relations of the kind

$$w^T y(t) = \text{constant} \tag{5.2}$$

hold, with w a k-vector with constant weights, we would like the numerical method to satisfy this relation as well. In case of constant emission terms represented by the vector Q, the equivalent of (5.2) is

$$w^T y(t) = \text{constant} + w^T Q \cdot t, \tag{5.3}$$

which should be satisfied by the numerical scheme as well. The special purpose solvers do not generally satisfy the conservation conditions exactly. As there exists a relationship between accuracy and conservation of mass, in Section 5.5.1 a technique will be described that improves the mass conservation and at the same time improves the accuracy. A simple example may illustrate this relationship. Consider the simple chemical reaction system

$$\begin{aligned} A &\xrightarrow{k_1} B, \\ B &\xrightarrow{k_2} A, \end{aligned} \tag{5.4}$$

with the corresponding differential equations

$$\frac{\partial A}{\partial t} = -k_1 A + k_2 B, \tag{5.5}$$

$$\frac{\partial B}{\partial t} = +k_1 A - k_2 B, \tag{5.6}$$

where A and B now denote the concentrations. A conservation relation holds for the system: $A(t) + B(t)$ is constant. This is obvious from the reaction set (5.4) and is mathematically reflected by the fact that

$$\frac{\partial A}{\partial t} + \frac{\partial B}{\partial t} = 0.$$

For $t \to \infty$ the solutions of A and B approach an equilibrium

$$[A, B] = (A_0 + B_0) \left[\frac{k_2}{k_1 + k_2}, \frac{k_1}{k_1 + k_2} \right],$$

where A_0 and B_0 are the initial values for A and B, respectively. Clearly, the equilibrium solution is directly dependent on the mass M in the system defined by $M{=}A + B$. If, for example, a numerical scheme systematically adds mass to the system, the exact equilibrium values for A and B will never be reached. They will converge to a different steady state. It is clear that in this example a close relationship exists between accuracy and conservation of mass. Though the present example is very simple, it may be considered to some extent representative for some reactions in ozone chemistry, see Section 2.3. The example makes clear why it may be worthwhile to pay special attention to mass conservation. Moreover, mass conservation together with positivity gives, for a large class of chemical kinetics problems, stability of the numerical scheme. Consider the mass relation

$$w^T y(t+\tau) \;\leq\; w^T y(t). \tag{5.7}$$

Often numerical methods that satisfy exact conservations relations (an "=" instead of a "$\leq$" in (5.7)) also satisfy the more general relation (5.7). In that case, positivity of the solution vector implies boundedness of the numerical solutions.

5.2 TWOSTEP

Recently, a special purpose solver has been developed by Verwer [65, 70]. Its description in this section is mainly based on [65, 70]. The solver is called TWOSTEP because it is based on the two step second order Backward Differentiation Formula (BDF). The general form of a BDF formula is given by

$$y^{n+1} \;=\; Y^n + \gamma\tau f(y^{n+1}), \quad \tau \;=\; t_{n+1} - t_n \tag{5.8}$$

where Y^n is an history vector, depending on the solutions from previous time steps and γ is a scalar variable. The step size is denoted by τ. In case variable step sizes are applied, Y^n and γ also depend on previous step sizes. In general purpose solvers like VODE the implicit relation defined by (5.8) is usually solved by using the (modified) Newton method. TWOSTEP however exploits the special form (5.1) of the chemical equations to rewrite (5.8) as

$$y^{n+1} = F(y^{n+1}) \;\equiv\; \left(I + \gamma\tau L^{n+1}\right)^{-1} \left(Y^n + \gamma\tau P^{n+1}\right), \tag{5.9}$$

to which Gauss-Seidel iteration is applied. Let $y^{(l)}$ denote the l-th iterate vector for y^{n+1}. Write the fixed-point form $y = F(y)$ in the componentwise form

$$y_i \;=\; F_i(y_1, \ldots, y_{i-1}, y_i, \ldots, y_k), \quad i = 1, \ldots, k. \tag{5.10}$$

Given an initial vector $y^{(0)}$, we compute for $l = 0, 1, \ldots, k$

$$y_i^{(l+1)} \;=\; F_i(y_1^{(l+1)}, \ldots, y_{i-1}^{(l+1)}, y_i^{(l)}, \ldots, y_k^{(l)}), \quad i = 1, \ldots k. \tag{5.11}$$

The iteration process (5.11) is explicit since L is a diagonal matrix. Replacing $y_i^{(l)}$ by $y_i^{(l+1)}$ in (5.11), would render the computation scalarly implicit. The two forms are identical if the P_i and L_i are independent from y_i for $i = 1, \ldots, k$. In general this is not true. In the reaction mechanism used in the model presented in this thesis, only one of the L_i depends on y_i.

Of course also Picard iteration may be applied, i.e.

$$y^{(l+1)} = F(y^{(l)}),$$

but numerical experiments have shown that Gauss-Seidel iteration is preferable in all test cases, see also [65]. For that reason we only consider Gauss-Seidel iteration in the numerical comparisons.

TWOSTEP applies the second-order BDF formula with variable step sizes which means that the history vector Y^n depends on the solution vectors y^{n-1} and y^n and on the step size ratio $c = (t_n - t_{n-1})/(t_{n+1} - t_n)$ according to

$$Y^n = \frac{1}{c(c+2)} \left[(c+1)^2 y^n - y^{n-1} \right]. \tag{5.12}$$

For the second order formula, the parameter γ is also depending on c and is given by

$$\gamma = \frac{c+1}{c+2}. \tag{5.13}$$

From the definition of the history vector Y^n it is seen that positivity of the solution values is not guaranteed. In fact all BDF methods with order higher than one may produce negative values. Only the first order BDF method, i.e. the Backward Euler method, guarantees nonnegative solution values. Negative values may give rise to instabilities in the integration process, so they have to be prevented. If during the Gauss-Seidel iteration a negative solution value is encountered, this value is cut off to zero. As starting vector $y^{(0)}$ for the iteration process we use the extrapolation

$$y^{(0)} = y^n + \frac{1}{c} \left(y^n - y^{n-1} \right). \tag{5.14}$$

Negative components of the initial vector are set to zero. The choice (5.14) ensures second order consistency, even if only one iteration is performed.

5.2.1 Conservation of mass

Suppose the exact solution of (5.1) satisfies

$$w^T y(t+\tau) = M, \quad M \text{ constant.}$$

It is then quite easy to prove that BDF methods are mass conserving, i.e. $w^T y^{n+1} = M$. To see this we write the general rth order BDF formula (5.8) in the from

$$\sum_{i=0}^{r} \beta_{n+1-i} y^{n+1-i} = \tau f(y^{n+1}). \tag{5.15}$$

Imposing the conservation relation yields, using $w^T f(y) = 0$ for all y,

$$\sum_{i=0}^{r} \beta_{n+1-i} w^T y^{n+1-i} = 0. \tag{5.16}$$

If we suppose that all past solution vectors y^{n+1-i} satisfy $w^T y^{n+1-i} = M$, it follows from $\sum_i \beta_i = 0$ that

$$w^T y^{n+1} = M. \tag{5.17}$$

Rosenbaum [50] proves that mass conservation not only holds for the exact solution y^{n+1} but also for the approximate solution obtained by any finite number of (modified) Newton iterations, provided that the analytical Jacobian is used. Under certain conditions this also holds if the Jacobian is approximated (see appendix D). However, when applying functional iteration according to (5.9)-(5.11), the approximate solution for y^{n+1} no longer satisfies the conservation relation. Only if the iteration process is continued until convergence, conservation of mass is achieved.

Now suppose there is a constant emission vector Q and the exact solution of (5.1) satisfies

$$w^T y(t+\tau) = w^T y^n + \tau Q.$$

Conservation for the rth order BDF formula can then be proved in a similar way as above. We then have $w^T f(y) = w^T Q$. Let the exact mass at time t_{n+1} be denoted by M^{n+1}. Again we assume that the previous time steps were mass conserving, i.e.

$$w^T y^{n+1-i} = M^{n+1} - (t_{n+1} - t_{n+1-i}) w^T Q, \quad i = 1, \ldots, \text{r}. \tag{5.18}$$

Imposing the conservation relation yields

$$\beta^{n+1} w^T y^{n+1} = -\sum_{i=1}^{r} \beta^{n+1-i} w^T y^{n+1-i} + \tau w^T Q,$$

which can be rewritten as

$$\begin{aligned} \beta^{n+1} w^T y^{n+1} = {} & \beta^{n+1} M^{n+1} + \\ & + \left(\sum_{i=1}^{r} \beta^{n+1-i} (t_{n+1} - t_{n+1-i}) + \tau \right) w^T Q \end{aligned} \tag{5.19}$$

using (5.16) and (5.18). The bracketed term in (5.19) is zero by definition. This can be seen from the order conditions. The relation (5.19) now reduces to

$$w^T y^{n+1} = M^{n+1},$$

which completes the proof.

5.2.2 The time stepping mechanism

In case variable step sizes are applied a time stepping mechanism is needed. For the selection of the step size the local error indicator E^{n+1} is used,

$$E^{n+1} = \frac{2}{c+1}\left(cy^{n+1} - (1+c)y^n + y^{n-1}\right), \tag{5.20}$$

which yields $\tau^2 y''(t_n) + O(\tau^3)$ upon substitution of the exact solution. The local error of the second order scheme is $O(\tau^3)$ so the present error indicator estimates the last Taylor term taken into account. Now consider the weighted error norm

$$\|E^{n+1}\|_w = \max(|E_j^{n+1}|/W_j^n), \quad j = 1, \ldots, k, \tag{5.21}$$

where $W_j^n = \text{ATOL}_j + \text{RTOL}_j y_j^n$ and ATOL_j and RTOL_j the absolute and relative error tolerance for component j. An integration step is accepted if $\|E^{n+1}\|_w \leq 1$ and rejected otherwise. In both cases the new step size τ_{new} is estimated by the common formula

$$\tau_{new} = \max(0.5, \min(2.0, \frac{0.8}{\sqrt{\|E^{n+1}\|_w}}))\tau. \tag{5.22}$$

The choice of the minimum growth factor of 0.5 in (5.22) is somewhat arbitrary, whereas the maximum growth factor of 2.0 is somewhat smaller than theoretically allowed. Stability results from [20] lead to

$$0 \leq \frac{\tau_{new}}{\tau} \leq 1 + \sqrt{2}. \tag{5.23}$$

The time step can further be restricted by a prescribed minimum and maximum value. In case of two successive rejections the integration process is simply restarted at t_n with initial value y^n.

5.2.3 The first steps

To apply the second order BDF formula, two values from previous time steps are needed. At the start of the integration only one is available, namely the initial value. Therefore the first step of the solver is carried out by a one-step formula. TWOSTEP performs a Backward Euler step (i.e. $Y^n = y_0$ and γ=1 in (5.9)) to start the integration process. Also the initial step size cannot be based on (5.22). Instead it is defined such that the first Taylor series term $\tau f(y_0)$ satisfies the accuracy requirement, according to

$$\tau = \min\left(\frac{W_j^0}{|f_j(y_0)|}\right), \quad j = 1, \ldots, k. \tag{5.24}$$

After the initial time step the two-step scheme is applied. The first step with the two-step scheme is performed with the same time step as the initial step. After that the time stepping mechanism is activated.

5.2.4 Variants

Following the same approach as in TWOSTEP, one may construct solvers based on higher order BDF formulae. To see whether this could be advantageous, we constructed 3STEP, based on the third order BDF formula. From the general form of the BDF methods (5.8) it can be seen that increasing the order to three only results in some extra overhead and bookkeeping. Evaluation of the history vector Y^n then is somewhat more expensive, since it now consists of a linear combination of three instead of two solution vectors from previous time steps. Also extra storage of one solution vector is required. In case of variable step sizes, the corresponding coefficients (including γ) depend on two time step ratios instead of one. This extra overhead has to be compensated for by the higher order resulting in taking larger time steps than TWOSTEP. In case of variable time steps, there is however a restriction that may restrain 3STEP from taking large time steps. The minimum and maximum growth factor for the time step are less favorable than for TWOSTEP, although theoretical bounds are not easily obtained. The higher the order r of the BDF method, the more step size ratios are involved in the stability analysis. Theoretical bounds for BDF methods with order $r \geq 3$ that *ensure* stability have been found by Grigorieff [20]. For BDF3 these bounds are

$$0.836 \leq \frac{\tau_{new}}{\tau} \leq 1.127 \,. \tag{5.25}$$

For order $r > 3$ these bounds are even less favorable. Fortunately, these bounds are unrealistic since they include all possible step size variations. Considering a constant step size ratio allows ratios between zero and 1.618, see [20]. In our implementation we allow ratios between 0.5 and 1.6, similar as for TWOSTEP. We did, however, apply a different way of computing the error estimate. The error is taken to be the difference between the BDF3 solution and the "virtual" BDF2 solution. This virtual solution is obtained by substituting the right hand side function at the advanced time level obtained by the BDF3 method into the BDF2 formula. This results into the expression

$$y^{n+1}_{\mathrm{BDF2}} = Y^n_{\mathrm{BDF2}} + \frac{\gamma_{\mathrm{BDF2}}}{\gamma_{\mathrm{BDF3}}}(y^{n+1}_{\mathrm{BDF3}} - Y^{n+1}_{\mathrm{BDF3}}).$$

Hence the error estimate is

$$E^{n+1} = y^{n+1}_{\mathrm{BDF3}} - y^{n+1}_{\mathrm{BDF2}},$$

which is $O(\tau^3)$. The definition of the weighted error norm is identical to (5.21). To obtain a new value for τ we also apply (5.22) but instead of taking the square root we now take the cube root of the weighted error norm.

The advantage of BDF methods over Runge-Kutta methods in general is that only one nonlinear system has to be solved per time step. A Singly Diagonally Implicit Runge-Kutta method of order 2 (SDIRK2) has already to solve two nonlinear systems per time step. Hence, the computational cost per time step for SDIRK2 is twice as much as for BDF2. An advantage of Runge-Kutta methods in general is that they may increase and decrease their

step size with an arbitrary factor. This property has to compensate the extra computational cost per time step. To see whether this could be advantageous we constructed an L-stable SDIRK2 method given by the following formulae

$$\begin{aligned} \zeta^{n+1} &= y^n + \tau\theta f(t_n + \theta\tau, \zeta^{n+1}), \\ y^{n+1} &= (1+\sqrt{2})\zeta^{n+1} - \sqrt{2}y^n + \tau\theta f(t_{n+1}, y^{n+1}), \end{aligned}$$

where $\theta{=}1-\frac{1}{2}\sqrt{2}$. Both stages can be solved using the Gauss-Seidel technique in exactly the same way as for TWOSTEP. The number of iterations may vary per stage. Some numerical experiments, however, revealed that a more efficient method than TWOSTEP is not obtained this way.

5.3 QSSA methods

A very popular method to solve (5.1) is the class of the so-called QSSA (quasi steady state approximation) methods, introduced by Hesstvedt, Høv and Isaksen in 1978, see [26, 27]. Although various variants exist, the underlying formula is the same for all variants. If the P_i and L_i in (5.1) are constant, this equation can be solved exactly,

$$y_i(t+\tau) = e^{-\tau L_i} y_i(t) + (I - e^{-\tau L_i}) L_i^{-1} P_i. \tag{5.26}$$

This suggests the associated integration scheme

$$y_i^{n+1} = e^{-\tau L_i(\tilde{y})} y_i^n + (1 - e^{-\tau L_i(\tilde{y})}) L_i^{-1}(\tilde{y}) P_i(\tilde{y}). \tag{5.27}$$

where the argument $\tilde{y}$ is still undefined. The argument is present because the P_i and L_i are not really constant but solution dependent. Note that the QSSA scheme is positive no matter how $L(\tilde{y})$ is evaluated as long as its argument is nonnegative.

5.3.1 First-order QSSA methods

The classical QSSA method

The classical, first order QSSA method is obtained if we select $\tilde{y}{=}y^n$ in (5.27), i.e.

$$y_i^{n+1} = e^{-\tau L_i(y^n)} y_i^n + (1 - e^{-\tau L_i(y^n)}) L_i^{-1}(y^n) P_i(y^n). \tag{5.28}$$

The method is fully explicit and that is what makes QSSA methods so attractive. The first order result follows from a Taylor expansion of (5.28). Substitution of the exact solution value $y(t_n)$ for y^n results into

$$y^{n+1} = y(t_n) + \tau\dot{y}(t_n) + \tau T(-\tau L)\dot{y}(t_n), \quad T(z) = \frac{e^z - z - 1}{z}. \tag{5.29}$$

From the Taylor expansion (5.29) we see that first-order consistency only holds for the non-stiff components for which $\tau \ll L_i^{-1}$ due to the fact that $T(z){=}O(z)$ for $z \to 0$. For the stiff components however we have $T(z) \sim 1$, so for these components a zero-order consistency holds. This provides an

example of local order reduction, see [15], Chapter 4. Of course, close to an equilibrium this reduction may not be felt. However, if a component is not close to equilibrium and yet $\tau \gg L^{-1}$, the QSSA scheme may introduce the steady state too quickly. This indicates that the accuracy can be low and unpredictable, to some extent, for large complicated chemical kinetics problems containing widely different time scales.

It can easily be shown that the scheme is not mass conserving. Rewriting (5.28) gives

$$y^{n+1} = y^n + (I - e^{-\tau L(y^n)})L^{-1}(y^n)f(y^n).$$

Imposing mass conservation yields

$$w^T(I - e^{-\tau L(y^n)})L^{-1}(y^n) = \tau w^T,$$

which is not fulfilled in general. In fact, mass conservation is only satisfied if the exponential e^z is approximated by $1 + z$. Substitution of this approximation results into the Forward Euler method, an inappropriate method to solve stiff systems because of its severe stability restriction.

In the original paper by Hesstvedt [26] the integration formula (5.28) is not used for all species. A simple Forward Euler step is performed by replacing the exponential by $1 - \tau L_i$ if $\tau L_i < 0.01$ and formula (5.27) is applied if $0.01 \leq \tau L_i \leq 10$. If $\tau L_i > 10$ the exponential is replaced by zero and the species is assumed to be in quasi steady-state, i.e.

$$y_i^{n+1} = \frac{P_i}{L_i}.$$

We note that these replacements of the exponential function are not essential. They are just used for efficiency reasons. However, in our experience using the exact exponential or the very accurate approximation (5.35) gives more accuracy.

The implicit counterpart of the classical QSSA scheme is obtained by using $\tilde{y}=y^{n+1}$. Functional iteration is applied to approximate y^{n+1} starting with the initial iterate y^n. This solver is used in the EMEP model [57, 56], where the following classification is made: for $\tau L < 0.01$ the Forward Euler method is applied and for $\tau L > 10$ the exponential formula is replaced by zero, resulting in the steady-state formula evaluated in y^n. For all other components the implicit formula is really used and some iterations are performed to obtain the corresponding solution values. In [70] this scheme is tested against the two-stage scheme proposed in [69]. The two-stage scheme is clearly superior to the EMEP scheme, so the EMEP scheme is not considered further. The two-stage scheme from [69] may be considered as a special case of the midpoint QSSA scheme that will be described in Section 5.3.2.

5.3.2 Second-order QSSA methods

A midpoint scheme

If we choose $\tilde{y}=\frac{1}{2}(y^n + y^{n+1})$ in (5.27), the method is implicit and of second order. The solution of the implicit equation is then obtained iteratively. A

straightforward Taylor expansion results into

$$y^{n+1} = y(t_n) + \tau y'(t_n) + \frac{\tau^2}{2} y''(t_n) + \tau T(-\tau L) f(\tilde{y}), \tag{5.30}$$

where $T(z)$ is given by

$$T(z) = -\frac{2(1-e^z)+z(1+e^z)}{z(1+e^z)}. \tag{5.31}$$

From the definition of $T(z)$ we see that for small values of τL we have $\tau T(-\tau L)$ is $O(\tau^3)$ as it should. Unfortunately, similar as for the first-order scheme from Section 5.3.1, for the stiff components order reduction occurs due to the fact that then $T(z) \sim 1$.

Second order may also be obtained by replacing P and L in (5.27) by $\tilde{P}$ and $\tilde{L}$ defined by

$$\tilde{P} = \frac{1}{2}[P(y^n) + P(y^{n+1})], \tag{5.32}$$

$$\tilde{L} = \frac{1}{2}[L(y^n) + L(y^{n+1})]. \tag{5.33}$$

Since no significant differences were observed, we only proceed with the second order version of (5.27) with $\tilde{y}=\frac{1}{2}(y^n + y^{n+1})$. Rewriting the scheme (dropping the argument $\tilde{y}$) in the form

$$y^{n+1} = y^n + 2(I + e^{-\tau L})^{-1}(I - e^{-\tau L})L^{-1} f,$$

and imposing conservation of mass results into the condition

$$2w^T(I + e^{-\tau L})^{-1}(I - e^{-\tau L})L^{-1} = \tau w^T.$$

Again we see that the conservation condition is violated. The only approximation $R(z)$ for e^z that makes the second order scheme conservative is

$$R(z) = \frac{2+z}{2-z}. \tag{5.34}$$

The approximation (5.34) is recognized as the second order diagonal Padé approximation which yields upon substitution the implicit midpoint rule. Unfortunately, this approximation may give negative solutions for $\tau L > 2$, though the resulting scheme is A-stable. Using this approximation would therefore be very inefficient.

Instead of the exact exponential a very accurate approximation $R(z)$ may be used for efficiency reasons only:

$$R(z) = \left[1 - z + \frac{1}{2}z^2 - \ldots + \frac{1}{720}z^6\right]^{-1}. \tag{5.35}$$

No significant differences were observed using this approximation instead of the exact exponential, whereas using the approximation, when implemented efficiently, saves approximately 20% computation time. Note that this approximation maintains the positivity of the solution.

The scheme with $\tilde{y} = \frac{1}{2}(y^n + y^{n+1})$ can be written in the efficient form

$$\tilde{y} = \frac{1}{2}(1+e^{-\tau L(\tilde{y})})y^n + \frac{1}{2}(I-e^{-\tau L(\tilde{y})})L^{-1}(\tilde{y})P(\tilde{y}). \tag{5.36}$$

This form (5.36) is completely in terms of $\tilde{y}$ which is convenient for coding the iteration process. As no significant differences were observed with Picard iteration, we use Gauss-Seidel iteration to approximate $\tilde{y}$, similar as in TWOSTEP (see below). The starting vector for the iteration process is obtained by extrapolation using y^{n-1} and y^n. Also the time stepping mechanism is the same as for TWOSTEP.

The method of Young and Boris

The hybrid method of Young and Boris is a predictor-corrector algorithm that also uses a classification of the species based on their loss rates and the time step. If $\tau L_i < 1$ the following predictor-corrector pair is used

$$\zeta^{n+1} = y^n + \tau f(y^n), \tag{5.37}$$

$$y^{n+1} = y^n + \frac{\tau}{2}\left(f(y^n) + f(\zeta^{n+1})\right), \tag{5.38}$$

where ζ^{n+1} denotes the predictor and y^{n+1} the corrector which is taken as the final approximation. If $\tau L_i \geq 1$ the predictor ζ^{n+1} is obtained by applying (5.27) with $\tilde{y}=y^n$ and the second order diagonal Padé approximation (5.34) for the exponential. The corrector is obtained in the same way but with P replaced by $\tilde{P}$ according to (5.32) with ζ^{n+1} instead of y^{n+1}, and L replaced by $\tilde{L}$ defined as

$$\tilde{L} = \frac{1}{2}\left[L^{-1}(y^n) + L^{-1}(\zeta^{n+1})\right]^{-1}, \tag{5.39}$$

and not by (5.33). The different definition of $\tilde{L}$ is not essential. Numerical tests have shown quite similar results for (5.33) and (5.39).

Again we note that there is no need to make distinction between different values of τL_i for other reasons than computational efficiency. If no distinction is made for different values of τL_i and all components are integrated as if $\tau L_i \geq 1$, the scheme of Young and Boris is quite similar to the QSSA scheme proposed by Verwer & Van Loon [69] in a comparative study of different QSSA schemes and some state-of-the-art solvers from the stiff ODE field. Apart from a different definition of $\tilde{L}$, the only difference then is the choice of the approximation of the exponential. Whereas Young and Boris use the second order diagonal Padé approximation, in [69] the second order subdiagonal approximation

$$R(z) = \left[1 + z + \frac{1}{2}z^2\right]^{-1} \tag{5.40}$$

is used. This choice is preferred because it nicely mimics the damping for e^{-z} for $z \to \infty$ and also guarantees nonnegativity. The second order diagonal approximation lacks these nice properties. Verwer & Van Loon [69] found that the predictor-corrector scheme using (5.40) performs notably better than the

scheme using (5.34). Note that the scheme defined by (5.36) becomes almost identical to the predictor-corrector scheme if only two Picard iterations are performed with y^n as starting vector for $\tilde{y}$. The only difference is the definition of $\tilde{P}$ and $\tilde{L}$. As already mentioned, no significant differences were observed due to the slightly different definitions of $\tilde{P}$ and $\tilde{L}$.

Extrapolated QSSA

Several attempts have been made to improve the accuracy of QSSA methods by extrapolation techniques. In [33] the classical first order scheme (5.28) is extrapolated by Richardson extrapolation. To integrate from t_n to $t_n + \tau$ a full classical QSSA step with step size τ is performed, resulting into the approximation y_τ^{n+1} and two steps are performed with halved step size resulting into the approximation $y_{\frac{1}{2}\tau}^{n+1}$. As final approximation for $y(t_{n+1})$ is taken

$$y^{n+1} = 2y_{\frac{1}{2}\tau}^{n+1} - y_\tau^{n+1}. \tag{5.41}$$

Although three solution vectors have to be constructed, the production and loss terms have only to be computed twice, since the initial values for the first full and the first half step are the same. The same holds for the evaluation of the exponential, since $\exp(-\tau L_i) = [\exp(-\frac{1}{2}\tau L_i)]^2$. Note that in the present scheme no iteration is necessary, in contrast to the previously described second order QSSA scheme. Extrapolated QSSA, however, may produce negative values whereas the other QSSA methods are positive. Negative values are simply put to zero, similar as in TWOSTEP.

The implementation of this method is based on a code written by A. Sandu [51]. As local error indicator E^{n+1} the difference between $y_{\frac{1}{2}\tau}^{n+1}$ and y_τ^{n+1} is used

$$E^{n+1} = y_{\frac{1}{2}\tau}^{n+1} - y_\tau^{n+1} \tag{5.42}$$

The weighted error norm is defined by

$$\|E^{n+1}\|_w = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left(\frac{E_i^{n+1}}{\mathrm{ATOL}_i + \mathrm{RTOL}_i y_{i,\frac{1}{2}\tau}^{n+1}}\right)^2} \tag{5.43}$$

If $\|E^{n+1}\|_w$ is larger than one and the time step was larger than the minimum step size, the step is rejected, otherwise the step is accepted. The new step size is computed according to

$$\tau_{new} = \max(0.125, \min(fac, \frac{0.9\|E^n\|_w^{0.2}}{\|E^{n+1}\|_w^{0.35}})), \tag{5.44}$$

where $fac = 1$ if the previous step has been rejected, and $fac = 8$ otherwise. The powers 0.2 and 0.35 are suggested by Gustafsson, see [24], pp. 32-35 and the references cited therein.

5.4 Other solvers

As a result of the current interest in atmospheric problems, a lot of effort is put into developing fast and efficient special purpose solvers. Most of these solvers are just variants on known principles. It is beyond the scope of this work to discuss all these variants. An interesting possibility, worth mentioning here, is perhaps the implicit-explicit solver presented in [60]. Like in many QSSA methods, the solver in [60] uses the fact that not all species are stiff. The species are divided into two groups: a group of slow species and a group of fast species. The slow species are integrated using the second order explicit Adams-Bashforth formula whereas the fast (stiff) species are integrated with a fully implicit method. We think that this approach also combines very well with TWOSTEP, because the BDF formula requires the solution of the species at the advanced time level. The solution of the slow species is explicit and hence directly available in the iteration process for the fast species. However, for the present application we do not expect this approach to result into a large gain in efficiency.

5.5 Improving the iteration process

For the implicit solvers, described in this chapter, some form of functional iteration is applied to solve the resulting nonlinear system. It turns out that the iteration process can sometimes be improved drastically by exploiting special problem characteristics.

5.5.1 Lumping

In the chemistry literature, a popular approach is lumping [26, 27]. The basic idea of lumping is to define a 'new' species, being a linear combination of two or more species from the system, that is easier to integrate than the individual species it consists of (i.e. the values of P and L for this new species are much smaller and there is a weaker dependence on the other species). Each integration step the 'new' species is integrated together with all other species. At the end of the integration step, the 'new' species is used to re-evaluate one of its components. For our chemical model, specified in Section 2.3, two such new, lumped species are considered: NO_x=NO+NO_2 and O_x=NO_2+O_3. This lumping of NO_2 and NO into NO_x etc. underlies the assumption that the first two reactions from the chemical mechanism are in some sense dominant in the whole set of chemical transformations, because if we only consider reaction 1 and 2, we have

$$\frac{d}{dt}NO_x = 0, \quad \frac{d}{dt}O_x = 0, \tag{5.45}$$

showing that for these two reactions NO_x and O_x are conserved. Consequently, if the first two reactions are truly dominant in the whole system, NO_x and O_x are expected to vary relatively slowly. This, in turn, implies

that the integration of these quantities can be done accurately, so that correcting one of the grouped species will make sense.

For both lumped species a differential equation can be specified in the canonical form, i.e. with positive P and L. For NO_x we get

$$\begin{aligned}\frac{d}{dt}NO_x &= -k_3 \cdot NO_2 \cdot OH - 2k_4 \cdot (NO_2)^2 \cdot O_3 + k_{16}HNO_3 \\ &= -k_3 \cdot (NO_x - NO) \cdot OH - 2k_4 \cdot NO_2 \cdot O_3 \cdot (NO_x - NO) \\ &\quad + k_{16}HNO_3 \\ &= P_{NO_x} - L_{NO_x}NO_x,\end{aligned}$$

where

$$\begin{aligned}P_{NO_x} &= k_3 \cdot NO \cdot OH + 2k_4 NO_2 \cdot O3 \cdot NO + k_{16} \cdot HNO_3, \\ L_{NO_x} &= k_3 \cdot OH + 2k_4 \cdot NO2 \cdot O3.\end{aligned} \tag{5.46}$$

In the same way, a differential equation for O_x can be derived. We then arrive at the following production and loss terms:

$$\begin{aligned}P_{O_x} &= \sum a_i k_{i+5} VOC_i \cdot OH + k_3 \cdot OH \cdot O_3 \\ &\quad + 3k_4 \cdot (NO_2)^3 + k_5(1 - b_2)NO_2 + k_{16}HNO_3, \\ L_{O_x} &= k_3 \cdot OH + 3k_4 \cdot NO_2)^2 + k_5(1 - b_2).\end{aligned} \tag{5.47}$$

Emission terms can simply be added to the production terms, if necessary. In case deposition is included in the chemistry, rewriting these terms will result into a production term and a loss term. For example, if we have a deposition velocity d for NO_2, causing an extra term $-d \cdot NO_2$ in the differential equation for NO_2, we rewrite this as $d(NO - NO_x)$ and the production and loss terms for NO_x are adjusted according to

$$P_{NO_x} = P_{NO_x} + d \cdot NO, \quad L_{NO_x} = L_{NO_x} + d. \tag{5.48}$$

In TWOSTEP and the QSSA schemes these species are computed at the end of each iteration (or at the end of one time step for schemes that do not use iteration), with P and L evaluated at the solution generated by the most recent solution values and are then used to re-evaluate the largest component in each lumped species. Hence, if $NO_2 > NO$, then NO_2 is recomputed from NO_x, by simply putting $NO_2 = NO_x - NO$, otherwise NO is recomputed. In the same way O_3 or NO_2 is recalculated from O_x.

For TWOSTEP the lumping is meant to improve the convergence of the iteration process. In a box model test (Chapter 6) we will see that the lumpings introduced here indeed improve the iteration process in such a way that the true BDF solution is 'almost' recovered. For implicit QSSA schemes, the situation is entirely different. The lumping will not help the iteration process converge. This is due to the fact that the underlying formula, in contrast to the BDF methods, is not mass conserving. If the nonlinear system is solved exactly, the lumping relation will not hold and vice versa. Yet, it may be expected that lumping also improves the QSSA solution, because the

order reduction becomes visible for large values of L. The lumping decreases the maximum values of L in some sense.

For implicit methods based on mass conserving formulae which solve the nonlinear system exactly (i.e. very accurately), the lumping 'trick' is redundant.

5.5.2 The EBI method

In [25] the Euler Backward Iterative method (EBI) is introduced. Like in TWOSTEP an iterative procedure is applied to solve the nonlinear system. To accelerate the convergence of the iterative process, explicit solutions are applied for groups of species. A similar procedure is followed in [62] where the approach is called Remote Coupling Algorithm (RCA). An advantage of this procedure is that schemes can be constructed such that some of the conservation relations are satisfied. For example, if in our chemical model the nonlinear equations for NO, NO_2, NO_3^a and HNO_3 are solved keeping the other species fixed, conservation for nitrogen is achieved. We have examined whether this approach could be useful for application in TWOSTEP and 3STEP, but some experiments showed that lumping is more effective. Both approaches may be combined, but application of lumping after solving part of the nonlinear equations did not result into better results than obtained with lumping alone.

5.6 Linear analysis for TWOSTEP

From the descriptions of the solvers in this chapter we know that the two special purpose solvers, TWOSTEP and QSSA, are not mass conserving in general. For TWOSTEP this is due to the fact that only a few Gauss-Seidel iterations are performed. If the iteration process is continued until convergence, the solutions will be mass conserving. For schemes based on the QSSA formula this is not true. In this section some analysis for linear chemistry is presented for BDF methods with Gauss-Seidel iteration for solving the nonlinear equations. We thus consider linear systems of the form

$$\dot{y}(t) \;=\; Ay(t) \;+\; Q, \quad A \in R^{k\times k}, \tag{5.49}$$

where $Q \in R^k$ specifies the emissions. By definition, the matrix A has the following property

$$\begin{array}{ll} A_{ij} & \geq\; 0, \quad i \neq j, \\ A_{ij} & \leq\; 0, \quad i = j. \end{array} \tag{5.50}$$

We suppose that one or more mass conservation relations hold of the form

$$w^T y(t) \;=\; w^T y(t_0) \;+\; (t - t_0) w^T Q \quad \text{with} \quad w^T A = 0.$$

The BDF solution for linear chemistry is given by the matrix-vector equation

$$(I - \gamma\tau A) y^{n+1} \;=\; Y^n \;+\; \gamma\tau Q \tag{5.51}$$

with formal solution

$$y^{n+1} = (I - \gamma\tau A)^{-1}(Y^n + \gamma\tau Q). \tag{5.52}$$

From $w^T A = 0$ it follows that

$$w^T(I - \gamma\tau A)^{-1} = w^T. \tag{5.53}$$

This result will be useful in the analysis. Only one general result will be derived. Most of the analysis is restricted to the following linear reaction system

$$\begin{aligned} y_1 &\xrightarrow{k_1} y_2 \\ y_2 &\xrightarrow{k_2} y_1 \\ y_1 &\xrightarrow{k_3} y_3 \end{aligned} \tag{5.54}$$

with a constant source vector $Q=[Q_1, Q_2, 0]^T$. The matrix A is given by

$$A = \begin{pmatrix} -(k_1 + k_3) & k_2 & 0 \\ k_1 & -k_2 & 0 \\ k_3 & 0 & 0 \end{pmatrix}. \tag{5.55}$$

The k_i may take arbitrary nonnegative values but here we assume $k_1, k_2 \gg k_3$. This assumption seems reasonable because it corresponds to situations which often occur in practical applications, although then the equations are nonlinear. We can think of system (5.54) as a linearized NO_x chemistry with, for example $y_1 = NO_2$, $y_2 = NO$ and $y_3 = NO_3$. For this system we have the following conservation relation

$$\sum_{i=1}^{3} y_i(t+\tau) = \sum_{i=1}^{3} y_i(t) + \tau(Q_1 + Q_2). \tag{5.56}$$

Instead of computing the solution of (5.51) by inverting the matrix $M = I - \gamma\tau A$, Gauss-Seidel iteration is applied to approximately solve the linear equation (5.51). In the linear case, the Gauss-Seidel iteration involves splitting the matrix M into $M = M_1 - M_2$, with M_1 the lower triangular part of M (including the diagonal). Let $\zeta^{(l)}$ denote the l-th iterate for y^{n+1}. The iteration scheme for y^{n+1} then reads

$$M_1\zeta^{(l+1)} = Y^n + \gamma\tau Q + M_2\zeta^{(l)},$$

or

$$\zeta^{(l+1)} = (M_1)^{-1}(Y^n + \gamma\tau Q) + (M_1)^{-1}M_2\zeta^{(l)}. \tag{5.57}$$

Let the amplification matrix $(M_1)^{-1}M_2$ be denoted by G_1 and $(M_1)^{-1}$ by G_2. The solution of the recurrence relation can then be written as

$$\zeta^{(l)} = (I - G_1)^{-1}(I - G_1^l)G_2(Y^n + \gamma\tau Q) + G_1^l\zeta^{(0)}. \tag{5.58}$$

Letting $l \to \infty$ should result into the exact solution of the linear system. The exact solution is independent of the start vector $\zeta^{(0)}$ for the iteration process, and thus G_1^l should go to zero for $l \to \infty$. Supposing this is true, we find that

$$(M)^{-1} = (I - G_1)^{-1} G_2. \tag{5.59}$$

The expressions (5.58) and (5.59) can be used to consider different properties of the iteration process. In order to do this, we evaluate several situations. Most of the analysis is restricted to one time step. We also restrict ourselves to considering conservation errors. Instead of for BDF2 we will sometimes give results for the Backward Euler method (BDF1). First, we derive a general result for convergence of linear systems.

5.6.1 Convergence of the iteration process

For linear chemical systems, defined by (5.49) where the matrix A satisfies (5.50), convergence can be proved. This result is based on Theorem 6.16 in [2] on convergence for nonnegative matrix splittings, see [2] and the references cited therein. The splitting $M = M_1 - M_2$ is nonnegative if $(M_1)^{-1}$ exists and $G_1 = (M_1)^{-1} M_2$ is nonnegative. According to its definition, M_1 is a lower triangular matrix with strictly positive diagonal entries and nonpositive off-diagonal entries. Such a matrix is always invertible and its inverse is known to be nonnegative. Since M_2 is an upper triangular matrix with nonnegative entries, the nonnegativity of G_1 follows directly. Hence, the matrix splitting is nonnegative. Theorem 6.16 from [2] states that a nonnegative splitting is convergent if M is nonsingular and $M^{-1}M_2$ is nonnegative. Assuming that M is nonsingular for $\tau \in [0, \tau_{max}]$, it suffices to prove that $M^{-1}=(I-\gamma\tau A)^{-1}$ is nonnegative. This is true for all τ in $[0, \tau_{max}]$. The proof follows from a result in [68].

Proof: Take a vector $x > 0$ arbitrary and let

$$z(\tau) = (I - \gamma\tau A)^{-1} x.$$

We know that $z(\tau) > 0$ for τ sufficiently small. Now suppose this is not true for all $\tau \in [0, \tau_{max}]$ and let τ_0 be the first value for which one (or more) of the components of $z(\tau)$ becomes zero. Let this be the i-th component. Then $z_i(\tau_0)=0$ and $z_j(\tau_0) \geq 0$ for all $j \neq i$. This gives the contradiction

$$0 = z_i(\tau_0) = x_i + \gamma\tau \sum_{j=1}^{k} A_{ij} z_j(\tau_0) > 0,$$

since $x_i > 0$ and the off-diagonal entries of A are nonnegative. Hence such a τ_0 does not exist.□

The conclusion thus is that Gauss-Seidel iteration applied to (5.51) is convergent for all $\tau \in [0, \tau_{max}]$. If all eigenvalues of A are located in the left half-plane, the matrix M is nonsingular for all $\tau \geq 0$ and we then have $\tau_{max} = \infty$. In real chemistry problems, the eigenvalues seem always to be

real and usually negative. Only a few very small positive eigenvalues can be present (see e.g. [52]). Hence we may conclude that τ_{max} is relatively large and that for practical step sizes the Gauss-Seidel iteration is convergent.

5.6.2 Effect of not continuing the iteration process

To illustrate what happens to the mass balance if we do not continue the iteration until convergence, consider the linear system (5.49) and one BDF1 step with one iteration (i.e. $Y^n=y^n$ and $\gamma=1$). We take $Q=0$ and y^n as the initial iterate for y^{n+1}. The scheme can then be written as

$$M_1 y^{n+1} = (M_1 + \tau A)y^n, \quad \forall n \geq 0.$$

Multiplying both sides with w^T gives

$$w^T M_1 (y^{n+1} - y^n) = 0,$$

since $w^T A = 0$. This result reveals that a "disturbed" conservation relation holds

$$\tilde{w}^T y^n = \tilde{w}^T y_0, \quad \tilde{w}^T = w^T M_1.$$

From the definition of M_1 it easily follows that $\tilde{w}^T$ is a vector with positive weights. To see the effect of this disturbed mass relation, consider y_1 and y_2 and reaction 1 and 2 of the example problem with $Q_1 = Q_2 = 0$. The corresponding matrix A takes the form

$$A = \begin{pmatrix} -k_1 & k_2 \\ k_1 & -k_2 \end{pmatrix}$$

It is clear that the sum of y_1 and y_2 is constant for this example problem, so $w^T = [1, 1]$. However, $\tilde{w}^T = [1, 1+\tau k_2]$ for Gauss-Seidel iteration. The exact solution goes to an equilibrium for $t \to \infty$ such that $k_1 y_1(\infty) = k_2 y_2(\infty)$. Suppose that the numerical solution also goes to an equilibrium with the same ratio between y_1^∞ and y_2^∞. The conservation error is then given by

$$w^T(y^\infty - y^0) = \frac{\tau k_1 k_2}{k_1 + k_2 + \tau k_1 k_2} \left[\frac{k_2}{k_1} y_2^0 - y_1^0 \right]. \tag{5.60}$$

The absolute value of the conservation error is thus determined by the extent to which the initial solution is out of equilibrium and the size of the factor in front of the bracketed term in (5.60). This error may be quite large since the ratio k_2/k_1 may be large.

5.6.3 Evaluation of the example problem

situation 1a: equilibrium; one time step

In this example we only consider y_1 and y_2 (i.e. $k_3=0$) and do not take emission into account (i.e. $Q_1=Q_2=0$). The iteration process (5.57) to obtain y^{n+1} now reduces to

$$\zeta^{(l+1)} = G_2 Y^n + G_1 \zeta^{(l)}. \tag{5.61}$$

The exact solutions for y_1 and y_2 converge monotonely to their equilibrium values which satisfy $k_1 y_1 = k_2 y_2$ as $t \to \infty$. In case of Gauss Seidel iteration, the matrices G_1 and G_2 are given by

$$G_1 = \begin{bmatrix} 0 & \dfrac{\gamma\tau k_2}{1+\gamma\tau k_1} \\ 0 & \dfrac{(\gamma\tau)^2 k_1 k_2}{(1+\gamma\tau k_1)(1+\gamma\tau k_2)} \end{bmatrix}. \tag{5.62}$$

$$G_2 = \begin{bmatrix} \dfrac{1}{1+\gamma\tau k_1} & 0 \\ \dfrac{\gamma\tau k_1}{(1+\gamma\tau k_1)(1+\gamma\tau k_2)} & \dfrac{1}{1+\gamma\tau k_2} \end{bmatrix}, \tag{5.63}$$

From the definition of G_1 we see that $G_1^l \to 0$ for $l \to \infty$ independent from the values of τ, k_1 and k_2. This is due to the fact that the matrix A has nonpositive eigenvalues, 0 and $-(k_1+k_2)$, hence $(I-\tau A)^{-1}$ exist for all $\tau \geq 0$. Thus the iteration process converges for all nonnegative τ. Suppose we apply BDF1 with start vector $\zeta^{(0)} = y^n$. Then we can express the conservation error E_l in y^{n+1} after l iterations in terms of the last iterate $\zeta^{(l)}$ and the mass conserving solution $\zeta^{(\infty)}$

$$\zeta^{(\infty)} = (I - G_1)^{-1} G_2 y^n.$$

The conservation error E_l can then be written as

$$E_l = w^T \left(\zeta^{(\infty)} - \zeta^l\right) = w^T \left[(I - G_1)^{-1} G_1^l G_2 - G_1^l\right] y^n. \tag{5.64}$$

After some tedious manipulation this can be rewritten as

$$E_l = \left[\frac{\tau^2 k_1 k_2}{(1+\tau k_1)(1+\tau k_2)}\right]^l \left(\frac{k_2}{k_1} y_2^n - y_1^n\right). \tag{5.65}$$

The above expression shows that always a conservation error is made. Only in case the y^n is already the equilibrium solution, no conservation error is made. The expression also shows that the conservation error decreases for every iteration, but the convergence is slow for large values of τk_i. In practical applications τk_i may indeed be relatively large for stiff components, especially when large time steps are taken. In such situations it seems preferable to perform a few extra iterations. From (5.65) it can be seen that the order of the components is of importance. If y_2 is treated first, E_l would have been given by

$$E_l = \left[\frac{\tau^2 k_1 k_2}{(1+\tau k_1)(1+\tau k_2)}\right]^l \left(\frac{k_1}{k_2} y_1^n - y_2^n\right).$$

which is a factor k_1/k_2 larger than (5.65). Hence, if $k_1 \gg k_2$, it is better to first treat y_1. In general, this suggests that the stiff components should be the first ones, especially in case only a few iterations are performed.

situation 1b: equilibrium; infinite time steps

If we assume that the numerical solution also converges monotonely to an equilibrium, then (5.65) suggests accumulation of mass errors: either the sum of y_1 and y_2 grows or decreases monotonely and the solution converges to an equilibrium with a different mass. Numerical results for this example support this suggestion. We still consider BDF1 and the solution of the last time step as the starting vector for the Gauss-Seidel iteration. Applying only 1 iteration and an infinite number of time steps results into expression (5.60). It can be shown that the error E_l for l Gauss-Seidel iterations and an infinite number of time steps can be written as

$$E_l = \frac{\alpha_l k_1}{k_1 + k_2 + \alpha_l k_1} \left[\frac{k_2}{k_1} y_2^0 - y_1^0 \right]. \tag{5.66}$$

No general formula was found for the α_l. From (5.60) it can be seen that $\alpha_1 = \tau k_2$. Evaluating α_2 and α_3 gives

$$\begin{aligned} \alpha_2 &= \frac{\tau^3 k_1 k_2^2}{1 + \tau(k_1 + k_2) + 2\tau^2 k_1 k_2}, \\ \alpha_3 &= \frac{\tau^5 k_1^2 k_2^3}{1 + 2\tau(k_1 + k_2) + \tau^2(k_1 + k_2)^2 + \ldots.}, \end{aligned}$$

which suggests that the α_l are positive and of order $\tau^{2l-1} k_1^{l-1} k_2^l$. In that case the conservation error satisfies

$$|E_l| \leq \frac{\tau^{2l-1}(k_1 k_2)^l}{k_1 + k_2} \left| \frac{k_2}{k_1} y_2^0 - y_1^0 \right|. \tag{5.67}$$

Numerical experiments confirm the expressions for $\alpha_1, \alpha_2, \alpha_3$ and indicate that (5.67) holds indeed.

situation 2: emission

We still consider only y_1 and y_2 but now a constant emission vector $Q = [Q_1, Q_2]^T$ is taken into account. The contribution of one time step with BDF1 to the mass of system should be equal to $\tau w^T Q$. From expression (5.58) we see that this contribution after l iterations is given by

$$\tau w^T (I - G_1)^{-1} (I - G_1^l) G_2 Q \tag{5.68}$$

and since $w^T (I - G_1)^{-1} G_2 = w^T$ the error E_Q made in the contribution of the source term can be written as

$$E_Q = \tau w^T (I - G_1)^{-1} G_1^l G_2 Q. \tag{5.69}$$

Evaluating this term results into

$$E_Q = \left[\frac{\tau^2 k_1 k_2}{(1 + \tau k_1)(1 + \tau k_2)} \right]^{l-1} \left(\frac{\tau^2 k_2 Q_2}{1 + \tau k_2} + \frac{\tau^3 k_1 k_2 Q_1}{(1 + \tau k_1)(1 + \tau k_2)} \right). \tag{5.70}$$

As expected, E_Q goes to zero for $l \to \infty$. More important, however, is the fact that if only a few iterations are taken, every time step again the same mass error will be made. If only one iteration is performed the front term in brackets in (5.70) vanishes. In that case the error associated with Q_2 approaches τQ_2 if τk_2 becomes large. The error term associated with Q_1 only becomes large if both τk_1 and τk_2 become large, due to the order in which the components are treated in the Gauss-Seidel process. This suggests that components for which source terms are specified should be the first ones. This suggestion, however, may be in conflict with the earlier observation that it seems better to first treat the stiff components. From this point of view it seems advisable not only to take at least more than one iteration per time step especially in case large time steps are possible, but also to introduce a mechanism to reduce this systematic error. The lumping procedure, described in Section 5.5.1, is such a mechanism. For the present and previous situation, lumping is trivial, since the exact mass $y_1 + y_2$ is known. It is then very simple to correct either y_1 or y_2. For the next example this is less trivial and there we will show the effect of lumping.

situation 3: equilibrium with sources

We now consider the full reaction system (5.54), including source terms. In this situation an equilibrium solution for y_1 and y_2 exists independent of the initial values for y_1 and y_2. This equilibrium solution is given by

$$\begin{aligned} y_1^{eq} &= \frac{1}{k_3}(Q_1 + Q_2), \\ y_2^{eq} &= \frac{k_1}{k_2 k_3} Q_1 + \frac{k_1 + k_3}{k_2 k_3} Q_2. \end{aligned} \tag{5.71}$$

The conservation error E_l after l iterations and one time step can be written as

$$E_l = E_{l,0} + E_{l,Q}, \tag{5.72}$$

where $E_{l,Q}$ denotes the error term associated with the source term Q and $E_{l,0}$ denotes the error term associated with the initial concentrations. Evaluating these error terms gives similar expressions as (5.70) and (5.65).

$$\begin{aligned} E_{l,Q} &= \left[\frac{\tau^2 k_1 k_2}{(1+\tau k_1 + \tau k_3)(1+\tau k_2)}\right]^{l-1} \times \\ &\qquad \left(\frac{\tau^2 k_2 Q_2}{1+\tau k_2} + \frac{\tau^3 k_1 k_2 Q_1}{(1+\tau k_1 + \tau k_3)(1+\tau k_2)}\right), \end{aligned} \tag{5.73}$$

$$E_{l,0} = \left[\frac{\tau^2 k_1 k_2}{(1+\tau k_1 + \tau k_3)(1+\tau k_2)}\right]^{l} \left(y_1^0 - \frac{k_2(1+\tau k_3)}{k_1} y_2^0\right). \tag{5.74}$$

As expected, both errors go to zero for $l \to \infty$. If y_1 and y_2 are in the equilibrium (5.71), the conservation error is zero. In that case, substitution of the equilibrium solution in (5.74) results into $E_{l,0} = -E_{l,Q}$.

Now suppose we apply lumping to this system by considering the species $x = y_1 + y_2$. Since y_1 and y_2 do not depend on y_3, we evaluate x after y_1 and y_2 and then compute the new approximation for y_3. From the approximation of x we recompute y_1 by putting $y_1 = x - y_2$. After some tedious algebra we arrive at the following matrices G_1 and G_2

$$G_1 = \begin{bmatrix} 0 & -\alpha\dfrac{\tau k_2}{1+\tau k_3} & 0 \\ 0 & \alpha\tau k_2 & 0 \\ 0 & -\alpha\dfrac{\tau^2 k_2 k_3}{1+\tau k_3} & 0 \end{bmatrix}, \tag{5.75}$$

$$G_2 = \begin{bmatrix} \dfrac{1-\alpha}{1+\tau k_3} & \dfrac{\tau k_3}{(1+\tau k_2)(1+\tau k_3)} & 0 \\ \alpha & \dfrac{1}{1+\tau k_2} & 0 \\ \dfrac{\tau k_3}{1+\tau k_3}(1-\alpha) & \dfrac{\tau^2 k_2 k_3}{(1+\tau k_2)(1+\tau k_3)} & 1 \end{bmatrix}, \tag{5.76}$$

where

$$\alpha = \frac{\tau k_1}{(1+\tau k_1+\tau k_3)(1+\tau k_2)}.$$

Since $w^T G_1 = 0$ and $w^T G_2 = w^T$ all iterates are mass conserving, independent of the start vector. This is easily verified by multiplying both sides in (5.57) by w^T. If y_2 is recomputed from x we no longer have exact conservation, not even if we first update y_2 and then y_3 in the iteration process, though also in that case the mass error is reduced substantially. Reducing the mass error does not necessarily mean that the solution with lumping is more accurate than without lumping, but intuitively one expects this to be the case. Numerical experiments with real chemical systems have to confirm this expectation. In the next chapter, the effect of lumping will be shown for the chemical model presented in Section 2.3.

Although the above result with lumping has been obtained using a simple linear model, it is of interest for the nonlinear chemical model, presented in Section 2.3, as well. As mentioned earlier, we can think of the present 3×3 system as linearized NO_x chemistry. If we exclude reaction 16 and 17 from the system and only consider reaction 1-4, we in fact recover the present linear example problem. If the concentrations of O_3 and OH and reaction rate k_4 are kept constant when updating NO_2, NO, NO_3^a and HNO_3, the system becomes linear for these species.

5.7 Linear analysis for QSSA

In this section partly the same analysis as in the previous section is done for a few QSSA schemes: the first-order classical QSSA scheme, the second order

midpoint scheme and the second order extrapolated QSSA scheme. First a result on stability for the first two QSSA schemes is derived. Next, situation 1 from the previous section is evaluated for the three schemes.

5.7.1 Stability for QSSA schemes

The classical QSSA scheme from Section 5.3.1 reads

$$y_{n+1} = e^{-\tau L} y^n + (1 - e^{-\tau L}) L^{-1} P, \tag{5.77}$$

with P and L evaluated in y^n. This can be rewritten as

$$y^{n+1} = y^n + \tau \Gamma f(y^n). \tag{5.78}$$

where the diagonal matrix Γ is given by

$$\Gamma = \text{diag}(\gamma_i), \quad \gamma_i = \frac{1 - e^{-\tau L_i}}{\tau L_i}.$$

For linear systems, (5.78) reduces to

$$y^{n+1} = y^n + \tau \Gamma A y^n. \tag{5.79}$$

Supposing that for the exact solution of the linear system one or more conservation relations of the form

$$w^T y^{n+1} \leq w^T y^n$$

hold, it can be shown that for the classical QSSA scheme a disturbed conservation relation

$$\tilde{w}^T y^{n+1} \leq \tilde{w}^T y^n$$

holds. This vector $\tilde{w}$ follows from multiplication of both sides in (5.79) by $\tilde{w}^T$. Since $w^T A = 0$, the disturbed conservation relation holds if $\tilde{w}^T \Gamma = w^T$. Since all $\gamma_i > 0$, $\tilde{w}^T$ is a vector with nonnegative weights. The stability in the sense of the inequality $\tilde{w}^T y^{n+1} \leq \tilde{w}^T y^n$ follows from the positivity of the solution vectors. If the exponential is replaced by some approximation, the result is still valid if the resulting vector Γ has positive entries for all $\tau L_i \geq 0$ and the solutions generated by the scheme are nonnegative.

In the same way as for the classical QSSA scheme, the stability result for the midpoint QSSA scheme from Section 5.3.2 follows. The definition of the γ_i is different:

$$\gamma_i = \frac{2(1 - e^{-\tau L_i})}{\tau L_i (1 + e^{-\tau L_i})}.$$

Since the γ_i are positive, stability follows from the positivity of the solutions.

For the extrapolated QSSA scheme a similar result can not be derived. The solutions may become negative and moreover the γ_i found for this scheme may not all have the same sign. In the evaluation of situation 1 of the example problem, it will be shown that the scheme is not stable for all possible values of τL_i.

5.7.2 Example problem: Classical QSSA

Since the classical QSSA scheme is fully explicit, it can be written as

$$y^{n+1} = G(\tau)y^n.$$

In the linear case, the amplification matrix G does not depend on y^n.

situation 1a: equilibrium; one time step

For the present example, G is given by

$$\begin{bmatrix} 1-\alpha & \frac{k_2}{k_1}\alpha \\ \frac{k_1}{k_2}\beta & 1-\beta \end{bmatrix}, \tag{5.80}$$

where $\alpha = 1 - e^{-\tau k_1}$ and $\beta = 1 - e^{-\tau k_2}$. The eigenvalues of this matrix are 1 and $1-\alpha-\beta$. The modulus of the latter eigenvalue is bounded by 1. This is in accordance with the stability result in Section 5.7.1. One step with this scheme results into the conservation error E

$$E = \left[\beta - \frac{k_2}{k_1}\alpha\right]\left(\frac{k_1}{k_2}y_1^n - y_2^n\right). \tag{5.81}$$

The conservation error is only zero if e^{-z} is approximated by $1-z$. This is, however, undesirable since the resulting scheme is the Forward Euler method. For small values of τk_i the error E can be approximated by

$$E \approx \frac{1}{2}\tau^2 k_2(k_1 - k_2)\left(\frac{k_1}{k_2}y_1^n - y_2^n\right). \tag{5.82}$$

The second order behavior of the conservation error is in accordance with the fact that the classical QSSA scheme is first-order. For large values of τk_i a very different result is obtained. We then arrive at

$$E \approx \left(1 - \frac{k_2}{k_1}\right)\left(\frac{k_1}{k_2}y_1^n - y_2^n\right). \tag{5.83}$$

This expression shows that for large values of τk_i a large conservation error is made and the solution will be very inaccurate. This implies that, in order to get reasonable accuracies with QSSA schemes, the time step cannot be too large. It also suggests that lumping may improve the conservation, since a lumped species is often defined such that it has a smaller loss rate than the loss rates of the species it consists of.

situation 1b: equilibrium; infinite time steps

If an infinite number of time steps is performed, we arrive at the conservation error

$$E = \frac{k_1\beta - k_2\alpha}{k_1(\alpha+\beta)}\left(\frac{k_1}{k_2}y_1^0 - y_2^0\right).$$

This error is a factor $\alpha+\beta$ smaller than the error made in one time step (cf. expression (5.81)).

5.7.3 Example problem: The midpoint QSSA scheme

We consider the midpoint QSSA scheme from Section 5.3.2. The scheme is given by (5.77) with P and L evaluated in $\frac{1}{2}(y^{n+1}+y^n)$. One time step with one Gauss-Seidel iteration (with start vector y^n) results into the following expression for the conservation error E_1

$$E_1 = \left[\frac{k_2}{k_1}\alpha - \frac{1}{2}(2-\alpha)\beta\right]\left(\frac{k_1}{k_2}y_1^n - y_2^n\right), \tag{5.84}$$

where $\alpha = 1 - e^{-\tau k_1}$ and $\beta = 1 - e^{-\tau k_2}$. For small values of τk_i (5.84) we have that

$$E_1 \approx \frac{1}{2}(\tau k_2)^2\left(\frac{k_1}{k_2}y_1^n - y_2^n\right). \tag{5.85}$$

The conservation error for one iteration is $O(\tau^2)$ for $\tau \to 0$. This is not surprising, since with start vector y^n the value y_1^{n+1} is updated with the first-order classical QSSA method. Hence, the conservation error for one iteration should be $O(\tau^p)$ with $p \geq 2$. Note that if we use Picard iteration with start vector y^n, the result of the first iteration is identical to the classical explicit QSSA method. The situation is completely different if we let the number of iterations go to infinity. Instead of a zero conservation error, we find

$$E_\infty = \frac{2}{k_1(4-\alpha\beta)}\left[(k_2-k_1)\alpha\beta) + 2(k_1\beta - k_2\alpha)\right]\left(\frac{k_1}{k_2}y_1^0 - y_2^0\right). \tag{5.86}$$

This expression is only zero if e^z is approximated by the second order diagonal Padé approximation $(2+z)/(2-z)$. However, using the exact exponential, evaluation of the above expression for small values of τk_i yields

$$E_\infty = \frac{1}{12}\tau^3 k_2(k_1^2 - k_2^2)\left(\frac{k_1}{k_2}y_1^0 - y_2^0\right). \tag{5.87}$$

The above result shows that the conservation error for QSSA will not go to zero in general, but is of order τ^3. Note that the scheme itself is second order, so that the order of the conservation error should indeed be 3 or higher. Even stronger, the above result suggests that under certain conditions the conservation error may grow with an increasing number of iterations. The results for the conservation error derived here are based on approximations of the exponential and hence not valid in general. Numerical experiments for the present example, however, supported this suggestion.

For large values of τk_i we clearly see the order reduction of the present QSSA scheme. Letting $\tau k_i \to \infty$ in (5.86) results into

$$E_\infty = \frac{2}{3}(1 - \frac{k_2}{k_1})\left(\frac{k_1}{k_2}y_1^0 - y_2^0\right). \tag{5.88}$$

5.7.4 Example problem: Extrapolated QSSA

The underlying scheme for the extrapolated QSSA scheme from Section 5.3.2 is the first order classical QSSA scheme. For the extrapolated QSSA, one time step is given by

$$y^{n+1} = G_{ex}(\tau)y^n, \tag{5.89}$$

where the amplification matrix G_{ex} is expressed in terms of the amplification matrix G for the classical QSSA scheme

$$G_{ex}(\tau) = 2G^2(\tau/2) - G(\tau).$$

The absolute eigenvalues of the amplification matrix G_{ex} are now no longer bounded by 1. The eigenvalue λ different from one is given by $\lambda = 3-5x+2x^2$, where $x=\exp(-\frac{1}{2}\tau k_1)+\exp(-\frac{1}{2}\tau k_2)$. Hence, $x \in [0,2]$. For $x \in [0,\frac{1}{2})$, we have $\lambda > 1$. Thus for relatively large values of τk_i the extrapolated QSSA scheme is unstable.

Evaluation of E_1 for small values of τk_i results into

$$E_1 \approx \frac{1}{24}\tau^3 k_2(k_1^2 - k_2^2)\left(\frac{k_1}{k_2}y_1^n - y_2^n\right). \tag{5.90}$$

Again we see that the error for small values of τk_i is not zero, but of $O(\tau^3)$. Expression (5.90) is exactly a factor of 2 smaller than the corresponding error for the midpoint QSSA scheme (5.87).

For large values of τk_i the situation is again entirely different. Letting $\tau k_i \to \infty$ gives the conservation error E_1

$$E_1 \approx \left(1 - \frac{k_2}{k_1}\right)\left(\frac{k_1}{k_2}y_1^n - y_2^n\right).$$

This error is 1.5 times larger than the corresponding error for the midpoint QSSA scheme and identical to the corresponding result obtained for the classical QSSA scheme.

Chapter 6

A Comparison for chemical solution methods

6.1 Introduction

Although the numerical stiff ODE field is well developed and an interesting variety of efficient and quite reliable stiff ODE solvers is available [24], the general experience is that for three-space dimensional transport-chemistry problems, where stiff ODE integrations are carried out at thousands of grid cells, still faster tailor-made solvers are needed. In Chapter 5 a number of such solvers has been described. In this chapter we compare TWOSTEP, 3STEP and extrapolated QSSA from the previous chapter with the state-of-the-art solver VODE from the numerical stiff ODE field. The main purpose of this comparison is to find out which method is best for application in the smog model. More precisely, we wish to find out whether explicit codes can be more efficient when applied in the smog model than the state-of-the-art solver VODE if we largely economize on the numerical algebra overhead of the modified Newton process by exploiting the sparsity of the Jacobian matrix. Since (large) fixed step sizes are attractive, the solvers are also tested with fixed step sizes. If it is possible to apply fixed step sizes in the real model, we have control over the total CPU time which we do not have otherwise.

To find an answer to this question, a box model test is presented. Naturally, this box model test has to simulate the real application of the solvers in the model as much as possible. Extensive comparisons for other box models and a wider variety of solvers can be found in [67] and in [52]. One of the conclusions in these publications is that there exists no 'best method'. Which method is most suited is highly problem-dependent. Roughly speaking, standard implicit solvers are more efficient if high precision is desired. High precision, however, is not required for the present application. In general, standard implicit solvers tend to become more efficient for large problems, provided that the numerical algebra overhead is kept at a minimum by exploiting the sparsity of the Jacobian matrix. For small problems (which may also be less stiff) probably tailor-made solvers are more efficient. Somewhere there is a break-even point, and in this chapter we will try to find its position for our chemical model.

6.2 Test methodology

6.2.1 The box model

The chemical model is identical to the one described in Section 2.3. This system is integrated over a period of 5 days, starting at 0:00h at day 1 and ending at 0:00h at day 5. Because we deal with ozone chemistry, we choose the meteorological conditions to be the conditions on a warm summer day. The daily variation of the temperature in Kelvin T_k and relative humidity rh are modeled as

$$\begin{aligned} T &= 293.1 - 1.91 * \sin\left(\frac{\pi}{12} tod\right) - 2.78 * \cos\left(\frac{\pi}{12} tod\right), \\ rh &= 0.6 + 0.0764 * \sin\left(\frac{\pi}{12} tod\right) + 0.114 * \cos\left(\frac{\pi}{12} tod\right), \end{aligned}$$

where *tod* denotes the time-of-day in hours. The solar angle is computed according to the formulae in Section 2.3 with latitude θ=52^o. In order to get realistic concentration profiles, emission and deposition terms have to be specified. In Table 6.1 these quantities are specified together with the initial values for the concentrations.

	Name	Species	emission	deposition	concentration
1	Sulphur dioxide	SO_2	$1 \cdot 10^6$	$3 \cdot 10^{-6}$	0.0
2	Sulphate aerosol	SO_4	0.0	0.0	0.0
3	Nitrogen dioxide	NO_2	0.0	$2 \cdot 10^{-6}$	$4.92 \cdot 10^{11}$
4	Nitrogen oxide	NO	$1.25 \cdot 10^6$	0.0	0.0
5	Ozone	O_3	0.0	$5 \cdot 10^{-6}$	$4.92 \cdot 10^{11}$
6	Hydroxyl radical	OH	0.0	0.0	0.0
7	Nitrate aerosol	NO_3^a	0.0	0.0	$2.46 \cdot 10^{11}$
8	Ethane	C_4H_6	$3.7 \cdot 10^5$	0.0	$2.46 \cdot 10^{11}$
9	Butane	C_4H_{10}	$1 \cdot 10^6$	0.0	$2.46 \cdot 10^{11}$
10	Ethene	C_4H_4	$2.9 \cdot 10^5$	0.0	$2.46 \cdot 10^{10}$
11	Propene	C_3H_6	$1.3 \cdot 10^5$	0.0	$2.46 \cdot 10^{10}$
12	Xylene	XYL	$3.3 \cdot 10^5$	0.0	$2.46 \cdot 10^{10}$
13	Isoprene	ISO	$1 \cdot 10^5$	0.0	$2.46 \cdot 10^{10}$
14	Carbon monoxide	CO	$2.5 \cdot 10^7$	0.0	$3.69 \cdot 10^{12}$
15	Nitric acid	HNO_3	0.0	$1 \cdot 10^{-5}$	0.0

Table 6.1: Emission and deposition values in [mlc/cm^3/s] and initial concentrations in [mlc/cm^3] for the box model.

6.2.2 Set up of experiments

The solvers are tested as if they were used in an operator splitting approach. This corresponds with the application in the smog model where the chemistry routine is called once per hour for all grid cells. We therefore split up the total

integration interval in 120 subintervals of 1 hour. For each subinterval we then restart the integration of the solvers. The total time interval is sufficiently long to include a number of diurnal cycles for the important photochemical transformations and to include a large set of different initial conditions due to the restarts. Frequently restarting a solver is not attractive since this involves an unusual amount of small steps in the start phase. In particular, this enlarges the overhead in the numerical linear algebra in stiff solvers that use the modified Newton process to solve nonlinear systems.

Our measure of accuracy is based on a modified relative root mean square error $RRMS_i$ for each species i, taken over the endpoints of all 1-hour intervals over the 120 hours. Hence,

$$RRMS_i = \sqrt{\frac{1}{|\mathcal{J}_i|} \sum_{n \in \mathcal{J}_i} \left(\frac{y_i^n - y_i(t_n)}{y_i(t_n)} \right)^2}, \tag{6.1}$$

where $\mathcal{J}_i = \{0 \leq n \leq N : \ |y_i(t_n)| \geq \text{ATOL}\}$, $N = 120$, $t_n = 3600n$ sec. and $y_i(t_n)$ represents a sufficiently accurate reference solution. The modification consists of the use of $\mathcal{J}_i$ and $|\mathcal{J}_i|$ in (6.1) in order to exclude solution values below ATOL from the norm computations. We then calculate the number of significant digits for the average of $RRMS_i$ over the k species, defined by

$$SDA = -\log_{10}\left(\frac{1}{k} \sum_{i=1}^{k} RRMS_i \right). \tag{6.2}$$

Our comparison focuses on modest accuracy, that is relative accuracies near 1%, since higher accuracy levels are redundant for the actual practice of three-dimensional air pollution modeling. The following set of error tolerances for the variable step size control will be used for all species

$$\text{ATOL} = 1, \quad \text{RTOL} = 10^{-l}, \quad l = 1, 2, 3, 4, 5. \tag{6.3}$$

Since the unit of concentration is number of molecules per cm^3, we therefore effectively invoke a relative error control. For some species (radicals) the concentration can be smaller than 1, but these values are insignificant for the overall solution and require no local error control. Since the four solvers use quite different solution techniques, and are therefore difficult to compare, efficiency is measured by CPU time (SGI-Indy workstation, f77 -O option, double precision).

6.3 Application of the solvers

All solvers use the ordering of the species as listed in Table 6.1, except of course for the version of VODE in which we reorder the species to reduce the number of fill-in elements in the LU decomposition of the Jacobian.

In case of variable step sizes, we prescribe a minimal and maximal step size for all solvers. Only VODE is also tested without bounds for the step sizes. These are, in seconds,

$$\tau_{min} = 1, \quad \tau_{max} = 900. \tag{6.4}$$

Step sizes below 1 sec. are redundant in this application. The minimal time constant values of importance for the actual practice are about 1 min. in size and species with a time constant smaller than 1 sec. almost instantaneously get in their (solution dependent) steady state when they are perturbed. Hence the choice of 1 sec. is reasonable and safe compared to 1 min. Note that without the lower bound of 1 sec. extremely small steps could be taken since atmospheric chemistry problems containing photochemical reactions can possess time constants as small as 10^{-9} sec, although the present box model is not that stiff. For VODE, the lower bound of 1 sec. thus serves to reduce the numerical algebra overhead stemming from the frequent restarts (matrix factorizations caused by step size changes). Finally, the maximal step size of 900 secs. is also quite reasonable on chemical grounds, although the solvers perform equally satisfactorily without this constraint.

6.3.1 TWOSTEP and 3STEP

The solver TWOSTEP, as described in [70] and in Chapter 5, has been applied in two different ways, in the remainder of this chapter indicated by TWOSTEP1 and TWOSTEP2. By TWOSTEP1 we mean the standard use, where at any time step two Gauss-Seidel iterations are used and the step sizes are constrained by (6.4) in case of variable step sizes. It should be emphasized that two iterations are by far not enough to let the Gauss-Seidel iteration fully converge. Our experience is that the overall accuracy of the code improves with the number of iterations, but the efficiency generally not. We therefore prefer a small number of iterations, which is attractive anyhow after a restart with a small step size. TWOSTEP2 refers to the same way of application, but in addition lumping is used to exploit special problem properties in order to obtain a more efficient numerical solution process. The lumping has been described before in Section 5.5.1.

The solver 3STEP, described in Section 5.2.4, is applied in the same way as TWOSTEP, denoted by 3STEP1 and 3STEP2. When applied with large, fixed time steps, TWOSTEP and 3STEP will also be applied with more than 2 Gauss-Seidel iterations.

6.3.2 QSSA

The QSSA solvers used in the experiments in this chapter are based on the extrapolated QSSA method based on [52], described in Section 5.3.2. In [43] we already compared the "mid-point QSSA" solver from Section 5.3.2 for almost the same box model test. We then concluded that TWOSTEP was superior. The solver from [52] is applied in two different ways, in the remainder of this chapter indicated by QSSA1 and QSSA2. As for TWOSTEP, QSSA1 refers to the standard use without any 'trick', whereas QSSA2 refers to the scheme to which in addition NO_x and O_x lumping is applied. The lumping is applied in the same way as in TWOSTEP. First all species are updated, next the production

and loss terms for the 'lumped species' are computed using the updated concentrations and the concentrations of the 'lumped species' are computed using the QSSA formula. Finally, the values of NO/NO_2 are updated using NO_x and NO_2/O_3 using O_x.

6.3.3 VODE

The solver VODE [12] has also been used in three different ways, indicated by VODE1, VODE2 and VODE3. To enable the step size restriction (6.4) we had to overrule the rejection strategy. Without this overruling the code returns with an error message due to the constraint $\tau_{min} = 1$ and interrupts the integration. In general this is perfectly all-right, of course, and VODE should not be blamed for it.

VODE1 is the standard black box use, i.e., no optional input is used and the method parameters ITASK (not essential for our comparison) and MF are set to 4 and 22, respectively. The choice MF=22 implies that the code generates the Jacobian automatically by numerical differencing, while the standard full matrix linear algebra routines DGEFA (factoring) and DGESL (backsolves) from LINPACK are used. It also implies extra storage because both the Jacobian and its LU-decomposed form are stored. This saves Jacobian updates, on the other hand additional storage may be a disadvantage for higher-space dimensional problems. Because no optional input is used, there is no constraint on the step size selection.

VODE2 denotes the solver as it is, but with the step size restriction (6.4) and provided with the analytical Jacobian (MF=21) instead of the numerical one. Like MF=22, the choice MF=21 means extra storage, since the Jacobian matrix is saved together with its LU decomposition. Usually, the Jacobian is overwritten with its LU decomposition. With the extra storage, the Jacobian may be kept constant during a number of time steps even if the time step changes significantly. In that case only a new LU decomposition needs to be formed.

VODE3 is completely identical to VODE2, except that now the sparsity of the Jacobian is exploited to reduce the costs of solving the linear systems in the modified Newton iteration. We emphasize that this can be very profitable for large systems, but less profitable for small systems like the present one. To keep the fill-in of the LU-factorization small, the components in the ODE system should be reordered. A natural way to try to achieve this is to reorder the species according to the number of nonzero elements in the corresponding row of the Jacobian. The species with the largest number of nonzero entries is put last and so on. Following this rule-of-thumb and manipulating a little, we succeeded in finding a reordering of the species for our chemical model that results in only one fill-in element in the LU decomposition. The ordering of the species used in VODE3 is: the VOCs, SO_2 and SO_4, NO, O_3, NO_2, HNO_3, NO_3^a and OH. We note in passing that one and the same sparsity structure is used for the whole time interval. At night, when photochemical reactions are switched off, the sparsity is somewhat larger, but for simplicity

we have not used this (small) advantage. For the given sparsity structure, the linear systems can be solved quite efficiently with the *ILU* (Incomplete *LU*) routines DSILUS (factorization) and DSLUI2 (backsolves) from the Sparse Linear Algebra Package (SLAP). SLAP is a public domain code written by Greenbaum and Seager (with contributions of several other authors) that is available from Netlib [16]. We should remark, however, that we have slightly modified the original SLAP versions to increase their efficiency. We specified the sparsity structures of the Newton matrix, L and U so that the SLAP routines do not need to compute these structures every time. The SLAP routines replace the LINPACK routines DGEFA and DGESL, respectively. Like the LINPACK routines, they factorize and backsolve, but omit all redundant calculations in which a zero occurs. It should be remarked, though, that now no longer pivoting occurs in the matrix factorization. This could give rise to errors in the linear system solution which otherwise would have been avoided. We have not experienced problems of this sort. Of course, if the factorization fails, then the step size control of VODE will detect this and a change in the step size will improve matters. It seems that for solving stiff ODEs pivoting is only rarely required (cf. [32, 53]). VODE3 thus has been prepared to solve the atmospheric chemistry problems with higher efficiency than VODE1 and VODE2.

6.4 Results for the box model test: variable step sizes

Figure 6.1 shows all accuracy-efficiency plots for the box model test. The marks on the plots correspond with the five different values for RTOL. For QSSA2 only the three smallest values for RTOL are visible in the plot. For RTOL=$10^{-1}, 10^{-2}$ the average number of significant digits were -2.37 and -0.93, respectively.

Recall that only modest accuracy (1%) is required. This corresponds with SDA=2 in the work-precision diagram. Hence, the lower left part of Figure 6.1 is the part of main interest.

6.4.1 Results for the special purpose solvers

As expected from earlier experiences (see [43, 67]), Figure 6.1 shows that lumping improves the accuracy of the solution considerably, both for the BDF solvers and QSSA. The BDF solvers are, however, clearly superior.

Experimentally we found that the TWOSTEP2 solution is in fact very close to the true BDF2 solution. The same holds for 3STEP2. Recall that only 2 Gauss-Seidel iterations have been carried out. So in this case the effect of lumping is considerable. Since the additional costs are relatively small, it is very attractive to use.

For large tolerances the CPU times for QSSA and the BDF solvers are comparable but the accuracy of QSSA is much lower. For smaller tolerances, the situation is even less favorable for QSSA: the accuracies are still lower

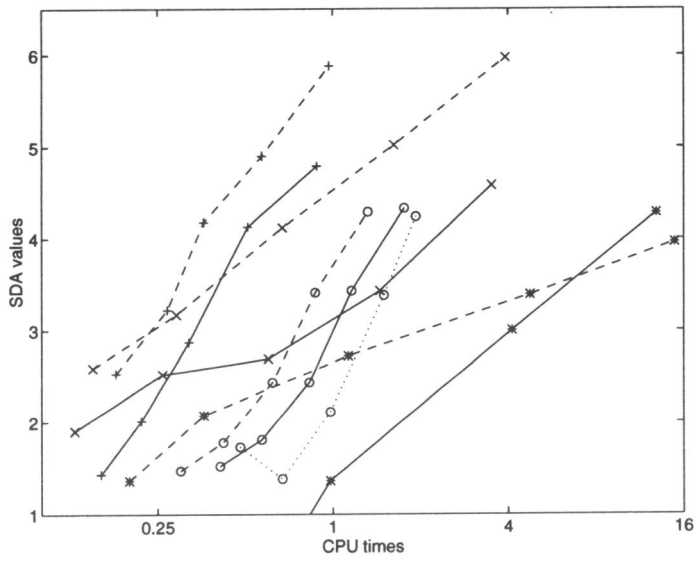


Figure 6.1: Work-precision diagram for the box model test: TWOSTEP1 (×,solid), TWOSTEP2 (×,dashed), 3STEP1 (+,solid), 3STEP2 (+,dashed), QSSA1 (*,solid) QSSA2 (*,dashed), VODE1(o,dotted), VODE2 (o,solid), VODE3 (o,dashed).

but the CPU times for QSSA are much larger. The latter is also illustrated by Table 6.2, where some integration statistics are given. The table shows that the number of steps taken by QSSA increases dramatically for decreasing tolerances. For the large tolerance values TWOSTEP2 and 3STEP2 perform comparably, but for smaller tolerances the higher order of 3STEP clearly pays off. Table 6.2 confirms this conclusion: the number of steps taken by 3STEP is much smaller than for TWOSTEP. This is also nicely illustrated by Figure 6.2. The line for 3STEP (not plotted) practically coincides with the line for VODE. This figure also shows the same behavior for all three solvers. Two peaks in the number of steps occur per day. The first peak occurs in the interval between 10:00h and 11:00h for all three solvers and the second in the interval between 19:00h and 20:00h for TWOSTEP2 and QSSA2 and sometimes one hour later for VODE2. At these peak hours TWOSTEP and QSSA take much more time steps than VODE and 3STEP. Especially QSSA takes a lot of time steps during these hours.

The peaks in the number of time steps per hour may be related to the change in the loss rates of the species. In Figure 6.3 the absolute change in loss rates per one-hour interval of NO_2 and NO is plotted. The numbers have been obtained by computing the absolute difference between L_{NO_2} and L_{NO} at the beginning and the end of the one-hour intervals. The observed peak in the number of time steps in the morning coincides with a strong change in the loss rate for NO and the peak in the afternoon coincides with

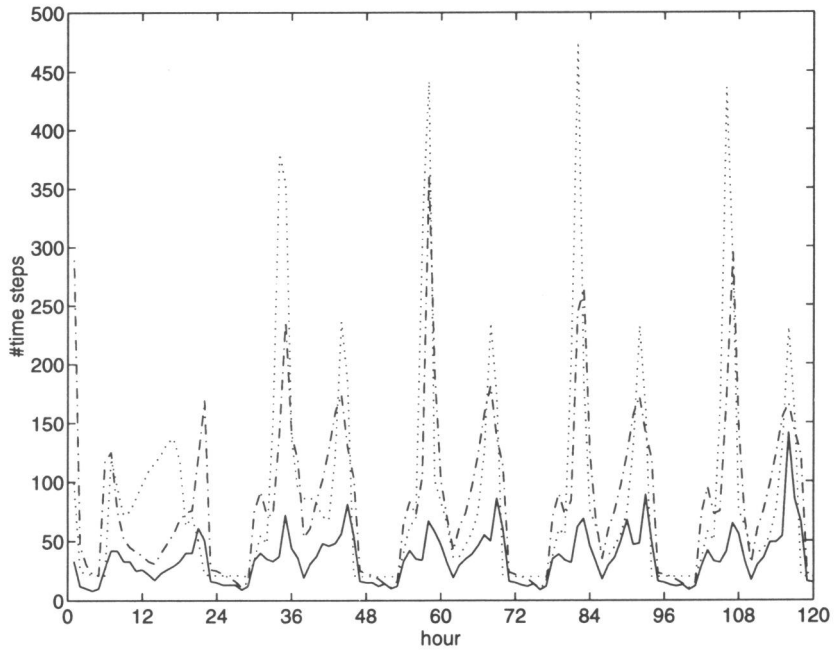


Figure 6.2: Number of steps per hour taken by the solvers for RTOL=10^{-3}: TWOSTEP2 (dash-dot), QSSA2 (dotted) and VODE2 (solid)

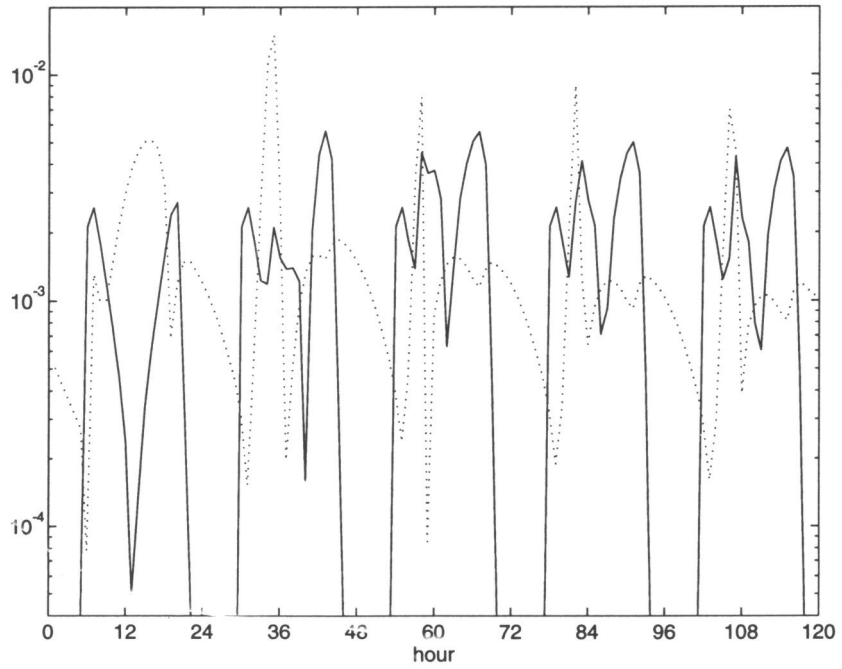


Figure 6.3: Absolute change in loss rate during the one-hour intervals for NO_2 (solid) and for NO (dotted).

a strong change in NO_2. This explains the increased number of steps taken by the solvers and especially why QSSA needs much more time steps than TWOSTEP in these interval, whereas both solvers are second-order accurate. If the loss rates vary relatively quickly within one time step, the QSSA scheme makes large errors, because the scheme underlies the assumption that the L_i (and in fact also the P_i) vary slowly within a time step. In such situations, the time stepping mechanism will prevent QSSA from taking large time steps.

The number of time steps taken by the codes is related to the relative error in some of the species. In the Figures 6.4 and 6.5 the relative errors in NO and O_3 are plotted together with the number of time steps taken by TWOSTEP1/2. The relative error for species i at time $t = t_n$ is computed by

$$\left| \frac{y_i^n - y_i(t_n)}{y_i(t_n)} \right|.$$

The behavior of the relative errors for NO and O_3 is quite similar for TWOSTEP1 and TWOSTEP2. The latter seems to give almost always smaller relative errors. This indicates a proper working of the lumping technique. As expected, the peaks in the number of time steps correspond with peaks in the relative errors for NO. This relation is not observed for the relative error in O_3 which has a different pattern from the relative error in NO. The errors in O_3 seem to behave more smoothly in time and the increases in the error do not seem to coincide with increases in the number of time steps. In the experiments with TWOSTEP1, the largest of the error estimates for the species was never the one for O_3. For RTOL=10^{-1} and 10^{-2} the largest error occurs for NO in 25-37% of the time steps and for OH in 30-36% of the time steps. In 8-10% of the time steps SO_2 causes the largest errors. For smaller tolerances, the percentages for NO and OH decrease. The errors in other species then also come into play. For example, for RTOL=10^{-5} the VOCs are for about 40% responsible for the largest errors. SO_2 then causes about 16% of the largest errors. The lumping in TWOSTEP2 hardly changes these percentages. The relatively large percentage for SO_2 cannot be prevented by lumping of SO_2 and SO_4 into SO_x. This is probably caused by the strong coupling between OH and SO_2. This coupling is stronger than the coupling between OH and the VOCs since the corresponding reaction rates are one order lower than reaction rate k_{14} (see Section 2.3).

6.4.2 Results for VODE

The restriction $\tau_{min} = 1$ improves the efficiency of VODE for the accuracy range considered. Implementing sparse matrix routines in VODE results in only a small gain in efficiency. The CPU times measured for VODE3 are about 25% less than for VODE2.

As expected, VODE2/3 outperforms VODE1. We found that this is due to the step size restriction (6.4) and not a result of using the exact Jacobian instead of a numerical approximation. For the present model, with only 15 components, the overhead of this numerical approximation is too small to become visible in the results.

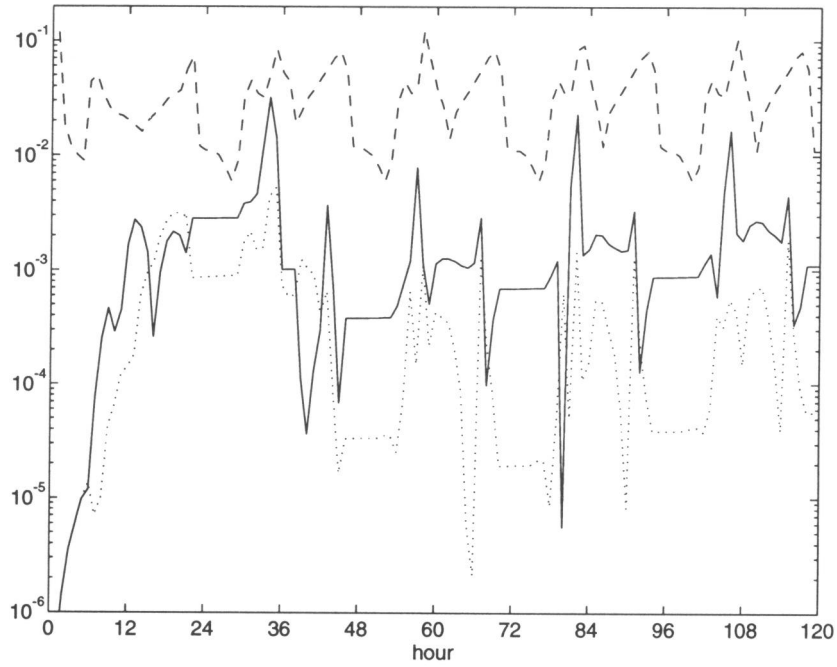


Figure 6.4: Relative error in NO for TWOSTEP1 (solid), TWOSTEP2 (dotted) with RTOL=10^{-2} and number of time steps per hour divided by 1000 (dashed)

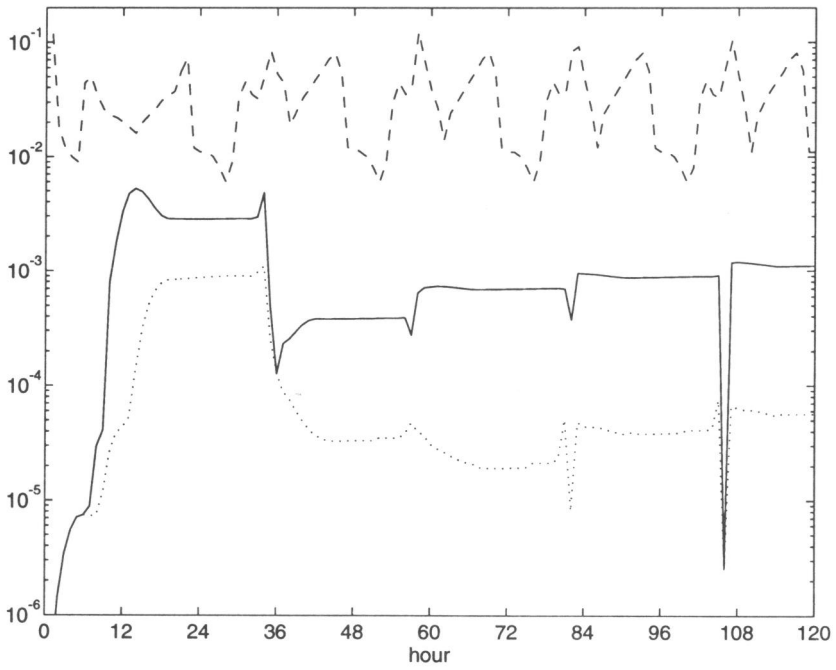


Figure 6.5: Relative error in O_3 for TWOSTEP1 (solid), TWOSTEP2 (dotted) with RTOL=10^{-2} and number of time steps per hour divided by 1000 (dashed)

method	RTOL 10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
vode1	1889 (34)	3007 (15)	4656 (61)	7318 (207)	9491 (297)
vode2	1752 (20)	2688 (7)	4201 (44)	6120 (69)	9869 (133)
vode3	1750 (19)	2688 (7)	4201 (45)	6113 (70)	9855 (134)
3step1	1975 (10)	2910 (13)	4453 (30)	7320 (19)	12978 (28)
3step2	1975 (9)	2910 (14)	4453 (30)	7321 (19)	12977 (28)
twostep1	2112 (0)	4257 (5)	10100 (5)	24504 (0)	59644 (1)
twostep2	2116 (0)	4252 (5)	10101 (5)	24503 (0)	59644 (1)
qssa1	2007 (133)	3607 (40)	10393 (1)	44081 (4)	138150 (3)
qssa2	1773 (11)	3248 (0)	10411 (1)	44095 (4)	137832 (3)

Table 6.2: Number of steps taken for the box model test with the number of rejected steps in brackets

Restriction (6.4) prevents VODE2 from taking very large step sizes which will reduce the accuracy at the end of the 1-hour intervals, but also prevents VODE2 from taking very small step sizes lower than 1 sec. in the initial phase. As noted before, these small step sizes are of no relevance for the accuracies we measure.

VODE2 spends only about 30% of the CPU time in routines that handle the *LU* decomposition and the backsolves, which of course is too small to get much gain in CPU by replacing VODE2 by VODE3. The latter needs approximately 20% less CPU time than VODE2. These numbers reveal that by using the sparse matrix routines, the linear algebra costs have been reduced by a factor 3.

Finally, when we compare with the most efficient VODE version, which is VODE3, we can conclude that TWOSTEP2 outperforms VODE3 convincingly. Also TWOSTEP1 is faster in the 1% error range.

6.5 Results for the box model test: fixed step sizes

If it is possible to use (large) fixed time steps, it would be very attractive to do so because one then has more control over the CPU time used for the chemistry in the true model calculations. In particular, if we can take large, fixed time steps of, say, a few hundred seconds, we fix the CPU time for the chemistry beforehand. Probably, the CPU time is also significantly smaller than in case of variable time steps with a minimum step size of one second. Keeping the amount of CPU time for the chemistry at a minimum is desirable, because the chemistry requires a relatively large amount of the total CPU time. The purpose of this section therefore is to find out whether large time steps are feasible for our chemical model and, if so, whether using large time steps is more efficient.

Several results indicate that taking large time steps may indeed be possible. In [70] it is shown that fixed time steps up to 900 seconds are possible for a much more complex chemical mechanism than ours. In [43] we showed that this is also possible for a slightly different box model test than the present one. The methods considered are only TWOSTEP2 and 3STEP2, because from the results for variable step sizes it is already clear that they are superior to QSSA. In [43] QSSA was also considered and it was found that the performance of TWOSTEP was much better. VODE is not included because this integrator was, in its present form, unable to integrate with large step sizes over the whole integration interval.

	TWOSTEP2					3STEP2				
τ	2	3	4	5	100	2	3	4	5	100
100	3.29	3.25	3.24	3.24	3.23	3.60	3.63	3.63	3.62	3.61
200	2.60	2.63	2.61	2.62	2.60	2.85	3.02	3.06	3.05	3.00
300	2.14	2.24	2.25	2.25	2.23	2.36	2.55	2.64	2.66	2.61
400	1.80	1.95	1.95	1.99	1.97	1.97	2.18	2.24	2.27	2.26
450	1.69	1.83	1.88	1.89	1.87	1.69	2.01	2.07	2.09	2.09
600	1.41	1.57	1.65	1.67	1.63	1.27	1.51	1.66	1.72	1.71
900	0.74	1.08	1.26	1.33	1.35	1.10	1.29	1.31	1.32	1.37

Table 6.3: SDA for fixed step sizes with 2,3,4,5 and 100 iterations

The results of the experiments are summarized in Table 6.3. Also results for 100 Gauss-Seidel iterations are listed because they are supposed to represent the accuracies of the true BDF solutions. The results clearly show that it is possible to use large time steps with only a few Gauss-Seidel iterations. To our surprise, the solutions are more than 1% accurate for time steps up to about 400 seconds. In the Figures 6.6 and 6.7 the relative errors in NO and NO_2 are plotted. The behavior of the 3 step sizes chosen is quite similar. The peaks in the error profiles occur at the same time as in Figure 6.4 for variable step sizes. Hence, the integration with fixed step sizes shows a 'normal' behavior. Again we conclude that taking fixed large time steps is possible for the present chemical model. From the Figures 6.6 and 6.7 we conclude that the error behavior for fixed time steps of 450 seconds is still acceptable, whereas in our opinion the errors for fixed time steps of 900 seconds become too large. The results with 5 iterations are more or less identical to the results with 100 iterations, indicating that 5 iterations are sufficient if we apply fixed time steps in the smog model. Note that the larger the time step, the smaller the differences between the results for TWOSTEP and 3STEP. Recall that the integration interval is split up into one-hour intervals, so that for example in case of fixed time steps of 900 seconds only 4 time steps are necessary per one-hour interval. Only two of them can be real BDF3 steps, since the first two steps are a BDF1 step followed by a BDF2 step.

In Table 6.4 the average time steps for the experiments with variable time steps together with the SDA values are given. The accuracies in this table

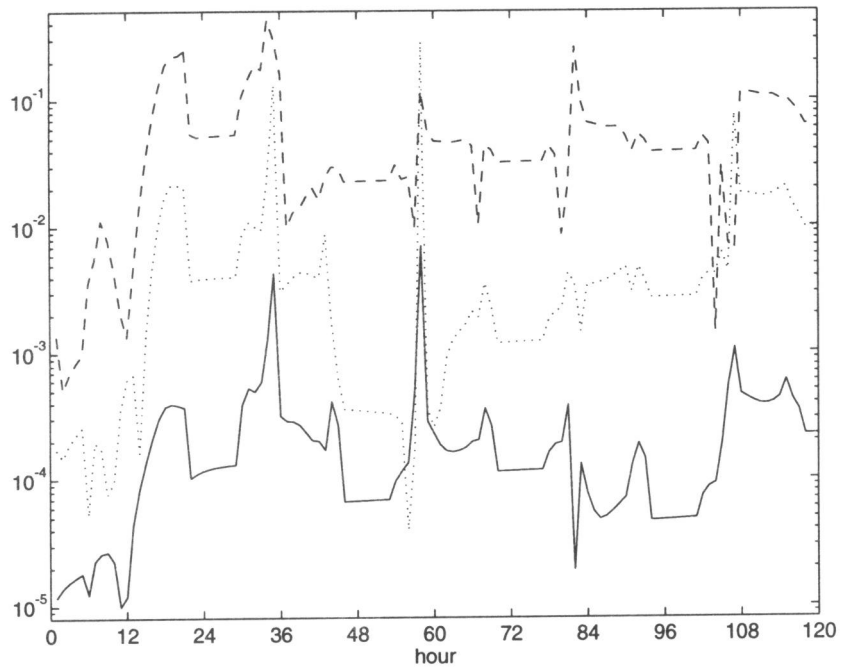


Figure 6.6: Relative error in NO for TWOSTEP2 with 3 iterations and fixed step sizes: 100 sec. (solid), 450 sec. (dotted) and 900 sec. (dashed)

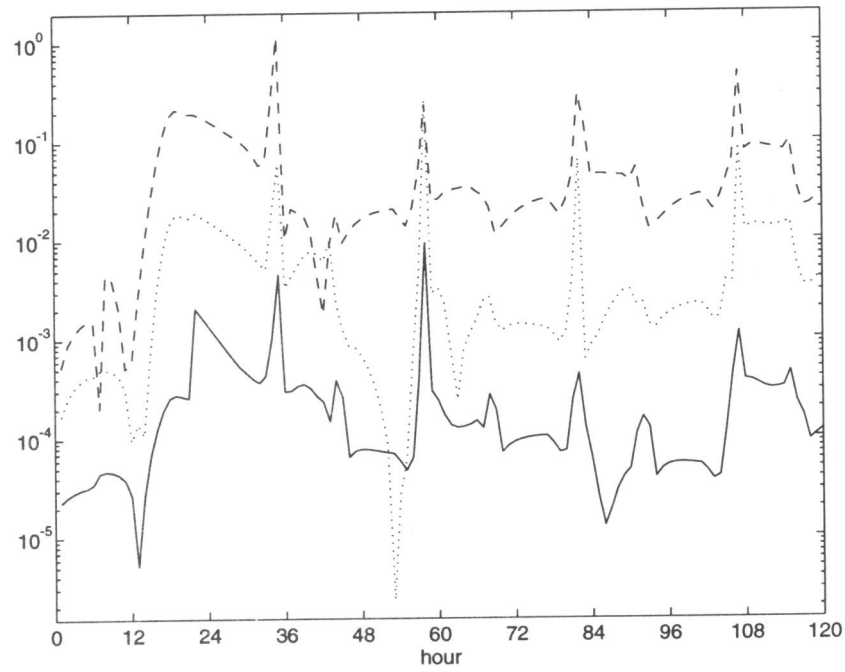


Figure 6.7: Relative error in NO_2 for TWOSTEP2 with 3 iterations and fixed step sizes: 100 sec. (solid), 450 sec. (dotted) and 900 sec. (dashed)

	TWOSTEP2		3STEP2	
RTOL	$\bar{\tau}$	SDA	$\bar{\tau}$	SDA
10^{-1}	204	2.59	219	2.53
10^{-2}	102	3.17	148	3.22
10^{-3}	43	4.12	97	4.18

Table 6.4: Average time steps $\bar{\tau}$ and SDA for the results with variable time steps from this section.

are quite comparable to the accuracies in Table 6.3 for about the same step size.

6.6 Further testing: general results

More experiments have been carried out. In [67] VODE and TWOSTEP are compared in the same way for two other chemical models with a slightly different measure for accuracy. In [52] more solvers and more test problems have been considered. In this section we give a few results from [67, 52] to illustrate the performance of explicit solvers against implicit solvers for other chemical models and/or other problems dimensions.

6.6.1 The methane CIRK chemistry

This chemical model was used in our comparison study [67]. For details we refer to [67] and the references cited therein. The model is used in long term, global studies and describes a methane oxidation cycle. It consists of 46 reactions between 19 species. Thirteen reactions depend on the solar zenith angle which is taken continuous and hence calculated in each time step. The problem is very stiff. Eigenvalues of the Jacobian lie between -10^9 sec^{-1} and -10^{-20} sec^{-1}, approximately. There are two extremely large eigenvalues which originate from the free radicals O^1D and O^3P. These species and hence these eigenvalues are absent in the chemical model presented in this thesis (Section 2.3), which explains the modest stiffness of that problem. The reordering of the species used by sparse VODE resulted in 12 fill-in elements in the LU decomposition. Thus the total number of nonzeroes after reordering is $111 + 12 = 123$.

Figure 6.8 shows all results obtained for the methane chemistry. First we notice that also for this problem the simple lumping trick improves the TWOSTEP accuracy considerably and for minor costs. The VODE results compare well with those for the chemical model used in the box model tests in this chapter. Supplying VODE with an analytical Jacobian and a minimal and maximal step size improves the performance significantly (VODE2). However, here also the gain in CPU from using the sparsity of the Jacobian is low, only 10%, similar as for the box model test presented in Section 6.4. In the accuracy region of greatest practical interest, both solvers perform well although TWOSTEP is the most efficient one.

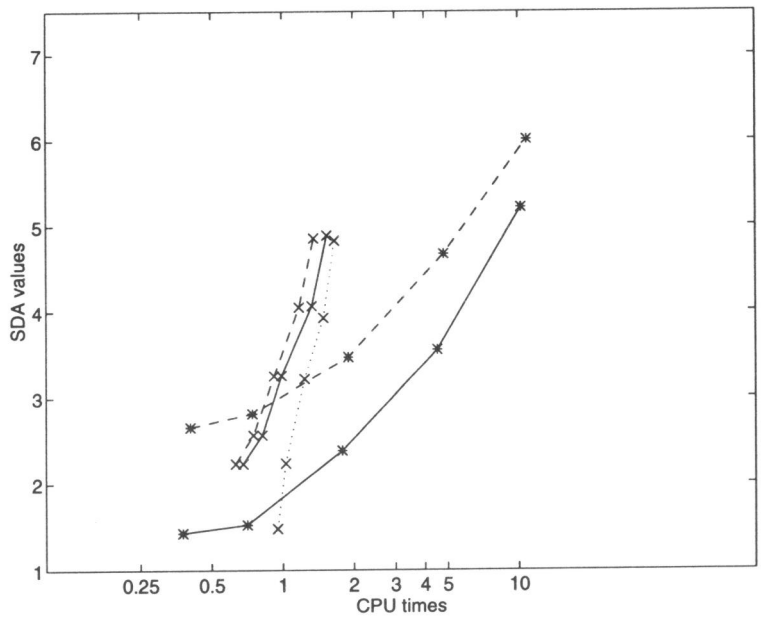


Figure 6.8: Results for the methane CIRK chemistry: TWOSTEP1 (*, solid), TWOSTEP2(*,dashed) VODE1 (x, dotted), VODE2 (x, solid), VODE3 (x, dashed).

6.6.2 The EMEP chemistry

In [67] also the EMEP chemistry has been considered. This chemical model is state-of-the-art in the field of regional air pollution modeling. The stiffness of the model is comparable to the stiffness of the CIRK model. However, the model is much larger. It consists of about 140 reactions between 66 species. For a detailed description of the model we refer to [57, 56].

The way of lumping applied in TWOSTEP2 now differs from the way of lumping described in Section 5.5.1. For TWOSTEP2 also two GS-iterations were used, but within each such iteration five group iterations on the $NO_y + O_3$ group are added (cf. [70]). The species in this group are strongly coupled, so it makes sense to perform this group iteration. We emphasize that this group iteration involves a minor change in the code and hence is very simply applicable. Because the group consists of only 7 species, the additional work is minor and it obviously improves the Gauss-Seidel iteration, as can be concluded from Figure 6.9 where the results for this box model tests are plotted.

The TWOSTEP2 result should be compared with the best result obtained for VODE, which clearly is the VODE3 case. We see that for the accuracy range of greatest practical interest, TWOSTEP2 and VODE3 are comparable. For higher accuracies the variable order VODE3 is more efficient because it then uses the higher order BDF formulas. The figure also nicely illustrates that by an intelligent use, standard stiff ODE codes like VODE can be improved

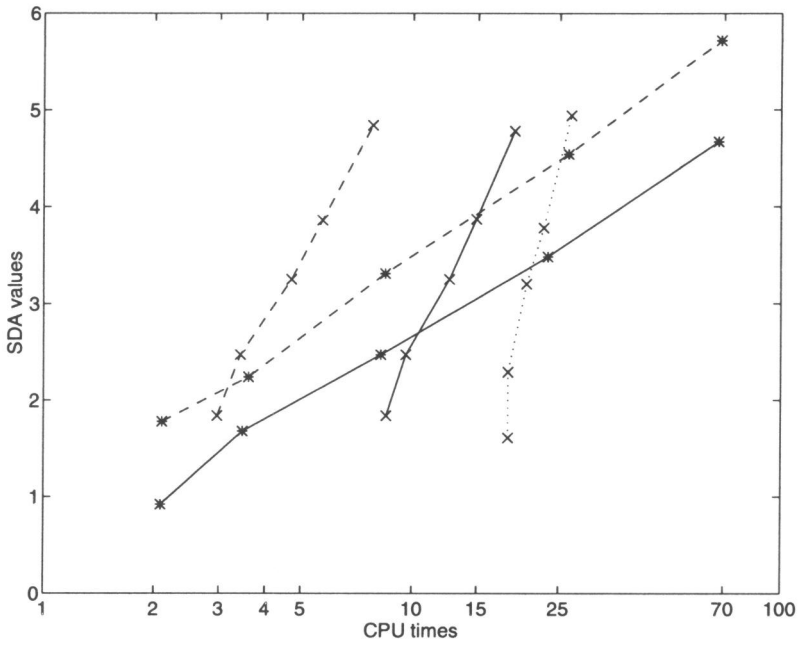


Figure 6.9: Results for the EMEP chemistry: TWOSTEP1 (*, solid), TWOSTEP2 (*, dashed), VODE1 (x, dotted), VODE2 (x, solid), VODE3 (x, dashed).

dramatically. In the low accuracy range VODE3 is about six times more efficient than VODE1. We emphasize that the difference between VODE2 and VODE3 is only due to the use of the sparse matrix techniques, which works out very well for this test problem due to its large number of components. The difference between VODE1 and VODE2 is due to using the analytical sparse Jacobian and the step size constraints (6.4). Both reduce part of the CPU time needed by the black box version VODE1.

6.6.3 Benchmarking Stiff ODE solvers

In [52] we considered a much wider variety of solvers and test problems. However, the special purpose solvers have been tested without any form of lumping. Since the performance of these solvers can sometimes be significantly improved by problem dependent modifications like lumping and group iteration, it is not quite fair to select a solver for a specific problem without considering a special purpose solver with one of such modifications.

Of interest in [52] is the conclusion that TWOSTEP is by far the best of the explicit solvers. Application of Gauss-Seidel iteration is in general more efficient than Jacobi iteration. The implicit solvers tested in [52], RODAS, SDIRK4 and VODE, have comparable performances, although their ranking relative to each other differs per problem. RODAS is the fastest in the 1%

error region for most problems. The implicit solvers tested in [52] have all been provided with sparse matrix routines. Since most of the test problems are large, this improves the performance of the solvers considerably.

For one test problem Gauss-Seidel and Jacobi iteration for solving the nonlinear system in TWOSTEP does not work for practical values for the time step. This test problem concerns a model that includes both gas-phase and liquid-phase chemistry, whereas in all other test problems only gas-phase chemistry is involved. For this test problem large eigenvalues λ_i of the Jacobian exist that do not correspond with the loss term L_i. For all other test problems the relation $\lambda_i \approx L_i$ does hold for the large eigenvalues. This should explain why for models including liquid-phase chemistry Gauss-Seidel iteration or Jacobi iteration does not work. We think however that by some form of lumping this problem can be overcome.

6.7 Concluding remarks

Considering our own box model tests and the results taken from [52, 67] we arrive at the following general conclusions.

- TWOSTEP is the most efficient (explicit) special purpose solver compared to the QSSA solvers tested here. In general there seems to be room for TWOSTEP as well as standard implicit solvers provided with sparse matrix routines. In the low accuracy region TWOSTEP always seems to be somewhat faster. Obviously, the (problem dependent) lumping technique and/or the group iteration are recommendable for TWOSTEP when only a few Gauss-Seidel iterations are used.

- An advantage of Gauss-Seidel iteration is that it works matrix free and hence the memory demand is low, which is of interest when grid vectorization is employed. As shown in [66], Gauss-Seidel iteration can be nearly optimally vectorized over the grid, in a similar way as modified Newton combined with sparse solution techniques in the code SMVGEAR [32].

  A further attractive feature of Gauss-Seidel iteration is that it can be efficiently extended to solve chemistry and vertical turbulent diffusion in a coupled way [66]. This is not true for the modified Newton process as regards the exploitation of sparsity. If diffusion is coupled with chemistry, then the sparsity of the chemistry Jacobian is almost completely lost in the factorization of the banded linear system.

- The sparse matrix technique based on the ILU routines from the SLAP library handles the solution of the linear systems well. We have encountered no difficulties in using VODE3, which solves the linear systems without pivoting. Similar experiences were reported by [32] and [53]. Other sparse matrix techniques may be more efficient, because the routines we used here and in [67] use indirect addressing.

- For large problems from atmospheric chemistry, like the EMEP model, the sparse matrix technique can lead to significant savings in CPU time for codes like VODE. This experience corresponds with the results reported by [32]. For atmospheric chemistry models of a more moderate size, the gain by exploiting sparsity hardly pays. For such models, with about 20 species say, the solution costs of the linear systems in VODE are simply too low compared to the costs of all other calculations.

Based on our own box model test presented in this chapter, the following more specific conclusions are drawn and recommendations are done.

- Explicit special purpose solvers provided with lumping seem to be the best choice. The results clearly show that TWOSTEP and 3STEP are the most efficient solvers for the present application and the accuracy range of interest.

- Lumping of NO_2 and NO into NO_x and of NO_2 and O_3 into O_x improves the iterative process substantially, so that only a few iterations have to be taken to arrive at an acceptable performance, with respect to efficiency, accuracy and conservation. For variable step sizes, 2 or 3 iterations are sufficient. For fixed, large step sizes, 3 to 5 iterations are sufficient.

- Implementing 3STEP2 with variable step sizes instead of TWOSTEP2 is expected to be somewhat more efficient (see Figure 6.1). Only one extra solution vector need to be stored. In the present implementation, this is an insignificant amount of memory. If the code is vectorized by looping over the grid cells within the chemistry routine, then an extra solution vector for each cell is necessary. The same holds for the reaction rates and the stoichiometry factors. Since the code of the model has not been written for vector machines, this argument plays no role. From this point of view, there is no reason not to implement 3STEP.

- It is possible to use TWOSTEP and 3STEP with large, fixed time steps. Up to time steps of about 450 seconds, the accuracies are acceptable. For TWOSTEP applied with 3 iterations and a step size of 450 seconds, the accuracy is 1%, approximately. From the figures with relative errors for NO and NO_2 we know that the relative errors in these species are most of the time smaller than the required 1%. Only for a few hours the relative error is larger than 1%, which has, by definition, a relative large influence on the SDA value. The same observation can be made for 3STEP.

- For fixed step sizes, 3STEP is a little more expensive than TWOSTEP. If we, however, consider the work precision diagram for both solvers, applied with fixed step sizes, the conclusion is that 3STEP is somewhat more efficient.

- In summary, our conclusion is that for the present application large, fixed steps should be taken. Experiments showed that this is possible from the accuracy point of view and it saves a considerable amount of CPU time compared to variable step sizes. This is important because the CPU time for the total model calculation should be restricted to a few hours. Since 3STEP seems to be more efficient than TWOSTEP2, we decided to implement 3STEP with fixed time steps of 450 seconds and 3 Gauss-Seidel iterations. Only for the first step, five iterations are performed.

Chapter 7

Model comparisons

In this chapter, results of model runs using meteorological data from both a winter smog and a summer smog episode will be presented. First we present results of model runs for the winter episode of November/December 1989. Next we present results for the summer smog episode of July 1989. For both episodes hourly measurements of relevant species in a number of stations in the Netherlands are available. Also measurements in EMEP stations are available, but only on daily averaged basis. Figure 7.1 shows the distribution

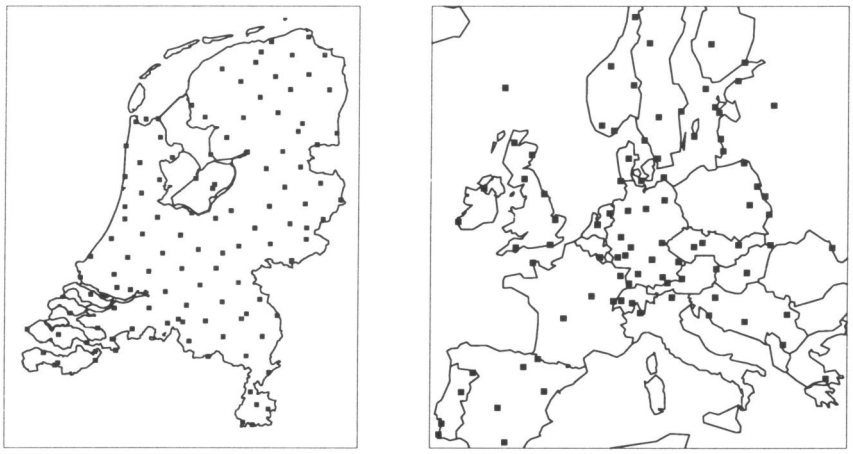


Figure 7.1: Stations in the Netherlands (l) and EMEP stations (r)

of the stations of the Dutch National Air Quality monitoring Network and the distribution of the EMEP stations over Europe. Unfortunately, for a large number of the EMEP stations no measurements are available or cannot be used because the altitude of the stations is too far above sea level. From the few remaining measurements a concentration distribution over Europe has to be derived. This will be a very crude estimation of the real distribution and comparison of the model output with measurements on a European scale will therefore only be indicative. For that reason we will restrict ourselves to comparisons with measurements of the Dutch monitoring network. Another reason to do so is that the model has been developed to give predictions for the Netherlands.

7.1 The November/December 1989 episode

In this section, results of experiments will be presented. First, a comparison is made between the results of model runs with the original model EUROS and CWIROS, in order to check whether CWIROS produces comparable results as EUROS. Next, the virtue of grid refinement is illustrated by comparing results obtained by using a number of refined grids to results obtained without using grid refinement. Finally a comparison will be made with measurements.

7.1.1 Comparison between EUROS and CWIROS

As the original model EUROS is a model for winter smog, both models are compared using data from the November/December 1989 smog episode. When running CWIROS without grid refinement, the differences between EUROS and CWIROS are:

- different advection schemes are used,
- CWIROS uses a much more complicated chemical model and consequently solves the resulting chemical kinetics problem in a different way,
- CWIROS uses other emission data than EUROS,
- CWIROS applies different deposition parameters for land and sea.

With respect to the chemical model, it should be noted that the more complicated chemical model in CWIROS reduces to a very simple model in wintry conditions when only considering SO_2 and SO_4 and involves the same reactions as the simpler chemical model in EUROS. Therefore, both chemical models are considered to be comparable when modeling SO_2 and SO_4 in winter. For this comparison however, we modeled the OH concentration in CWIROS in the same way as it is done in EUROS, where the OH concentration is prescribed as a function of time. This approach decouples SO_2 and SO_4 from the other species and thus simplifies the chemistry if only results for SO_2 and SO_4 are desired (see [42]).

The operational code EUROS is run every day simulating a period of 5 days (or 120 hours). Therefore we chose a five days period from the selected episode, from 27 November 1989 12:00 GMT till 1 December 12:00 GMT (for initialization purposes, the model starts one day earlier than the specified time). In Figure 7.2 solution plots of both model runs at 1 December 12:00 GMT are given. Apart from differences which are probably caused by the above mentioned reasons, both plots are in good agreement with each other. Both plots show an increased SO_2 level mainly in or close to the North-Eastern part of the Netherlands. This is in accordance with actual measurements, as we will see later on.

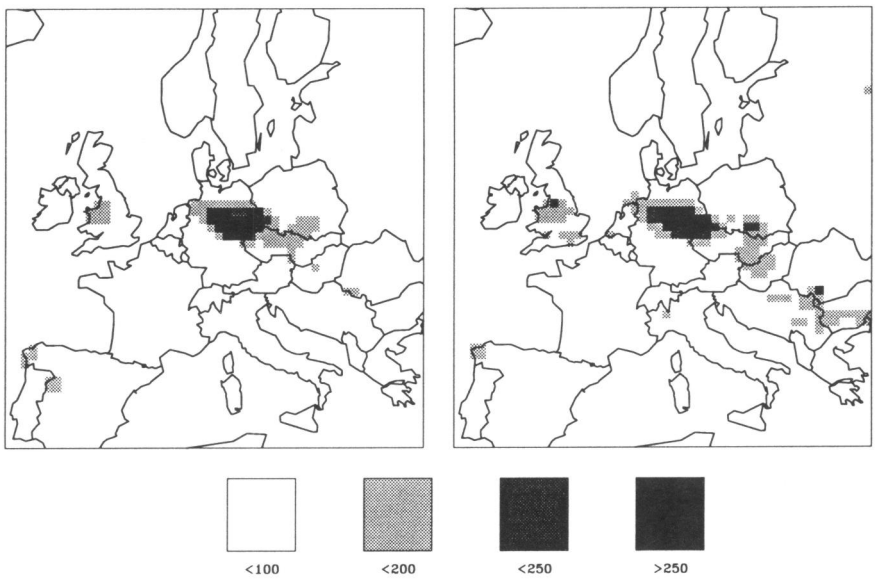


Figure 7.2: Solution plots for EUROS (l) and CWIROS (r) without grid refinement, SO_2 in $\mu g\, m^{-3}$ for 1 December 1989, 12:00 GMT.

7.1.2 Grid refinement

The model runs have also been performed using 2, 3 and 4 grid levels. On grid level 2 and 3 we always refined a rectangular area containing the Netherlands. Further, grid refinement is enforced around all sources with the restriction that grid refinement is never applied *outside* the region defined by $[0^o, 15^o] \times [-15^o, 0^o]$ to prevent unnecessary and time consuming grid refinement. In the solution plot the area in which refinement is allowed, is indicated by a rectangle. In Figure 7.3 the solutions of two model runs are plotted. Both solution plots are in good qualitative agreement with each other, indicating a proper functioning of the grid refinement. The plots also show that grid refinement results into a (slightly) different solution, as was to be expected. Furthermore, as in Figure 7.2, the solution plots show a cloud with polluted air over central and north-west Europe that just passes through the north-eastern part of the Netherlands. The latter observation is in accordance with Dutch measurements, as we will show in Section 7.1.3. Table 7.1 gives some statistical information about the four model runs. The numbers in the last column may serve as a measure for the efficiency of the refinement procedure. The numbers specify the percentage of grid cells used in the model runs relative to the number of cell necessary in case of a uniform fine grid on the maximum grid level used. In the model runs, refining only the Netherlands results into 196 cells on level 2, 616 cells on level 3. These numbers are relatively small compared to the average number of cells used

MAXLEV	TOL	grid level 1	2	3	4	total	uniform	%
1	-	2860	-	-	-	2860	-	-
2	0.5	2860	2084	-	-	4944	11440	43.2
3	0.5	2860	2054	4189	-	9103	45760	19.9
4	0.75	2860	1909	3323	7242	15334	183040	8.4

Table 7.1: Average number of cells for the model runs

on these grid levels, according to Table 7.1. If the Netherlands are further refined on level 4, it would take 1824 extra cells. However, we do not refine the Netherlands on level 4 automatically. If the Netherlands need further refiment the space monitor is supposed to take care of that. This prevents unnecessary refinement in (parts of) the Netherlands, and thus saves some computation time.

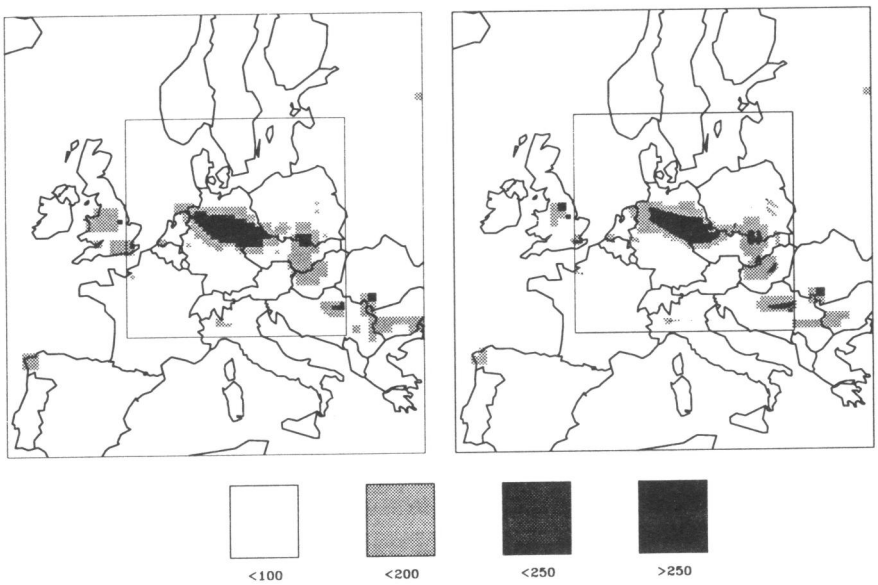


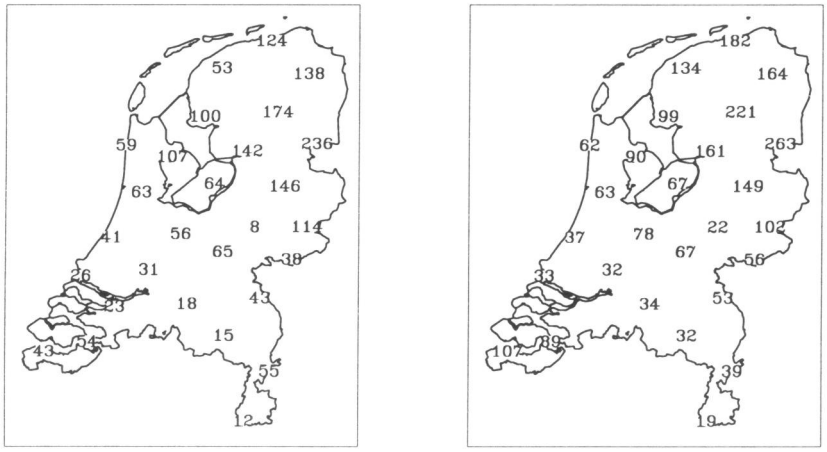
Figure 7.3: Solution plots for CWIROS using 2 grid levels (left) and 4 levels (right). SO_2 in $\mu g\, m^{-3}$ at 1 December 1989, 13:00 MET.

7.1.3 Comparison with Dutch measurements

Comparing model results with observed concentrations is difficult. Not only model errors are present, but we also have to deal with uncertainties in the input parameters. An important parameter in the present experiment is the r_c value (the surface resistance) of SO_2. This parameter influences the

dry deposition: the higher the surface resistance, the lower the deposition velocity. As SO_2 is not very reactive (it is only slowly transformed into SO_4 in our chemical model), dry deposition may cause significant removal of SO_2. However, the surface resistance of SO_2 is strongly dependent of the soil condition. In case of a frozen soil and snow this value is about five times higher than the default value 100 s/m. Because we do not have detailed information about the surface conditions in the selected period, an estimation for the surface resistance for SO_2 had to be made. Fortunately, temperature fields are available. We therefore estimated the surface resistance of SO_2 based on the local temperature. If the temperature is lower than $-1°C$ we take $r_c = 540$, if the temperature is higher than $1°C$ we take $r_c = 100$ and between $-1°C$ and $+1°C$ the surface resistance varies linearly between 540 and 100.

In Figure 7.4 measurements for the Netherlands are plotted. As can be



12:00 MET 13:00 MET

Figure 7.4: SO_2 measurements in $\mu g\, m^{-3}$ for 1-12-1989

seen from Figure 7.4, the highest concentrations are observed in the North-Eastern part of the Netherlands. From Figure 7.4 it can also be seen that a relatively large concentration gradient is present in the Netherlands, compared to the spatial resolution on the base grid. Model calculations on the base grid only will therefore not be able to reproduce this concentration gradient. However, we may expect that using a number of fine grid level improves the results especially near the gradient. Figure 7.5 shows the computed distributions above the Netherlands using no grid refinement and using four grid levels. Clearly the grid refinement delivers a much more realistic solution plot. It also nicely shows local sources that are invisible on the coarse grid solution. However, comparing Figure 7.5 with Figure 7.4 reveals that the computed concentration cloud in the northern part of the Netherlands

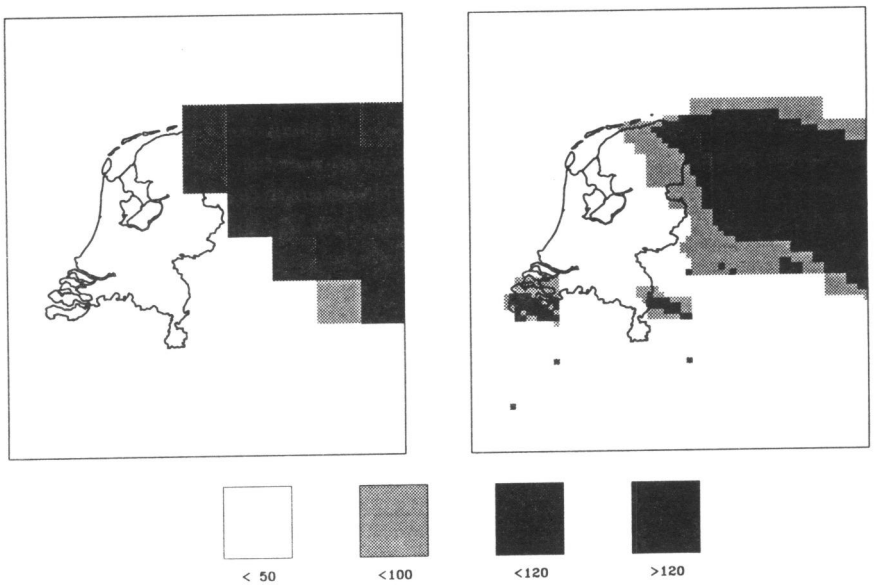


Figure 7.5: Solution plots for SO_2 in $\mu g\, m^{-3}$ at 1 December 1989, 13:00 MET. Left: 1 grid level; Right: 4 grid levels

has a somewhat different position than the observed cloud. This also seems to be the case with the small cloud in the south-west. Computed and observed concentrations in a measurement station will therefore not match, just because the computed concentration cloud may miss the station whereas in reality it just passes the station. In particular, this will happen for stations at the edge of the cloud.

The following will show that grid refinement sometimes gives less good agreement with observations. Possible explanations will be given. These explanations will reveal that in order to exploit the virtue of a higher numerical resolution, the model input and model coefficients become more critical. Our first comparison concerns three measurement stations from the Dutch *Air Quality Monitoring Network*, the stations 1-3 specified in Table 7.2. These three stations are located in the same grid cell on the base grid. Moreover, the three stations are very close to the observed concentration gradient. As can be seen from Table 7.2, station 3 is the closest one. Also in station 3 the highest SO_2 concentration is observed. Figure 7.6 shows the measured concentrations in the three stations. Figure 7.6a shows a peak in the observed concentrations on 1 December at 12:00 for station 3 and at 13:00 for the other two stations. Although the three stations are located in a small geographical area, the observed peak values are quite different. On the other hand, the time behavior for the three stations is comparable, so that one may expect a similar time behavior for the model calculations in the coarse grid cell. From Figure 7.6b it is seen that this is indeed the case and also that the model

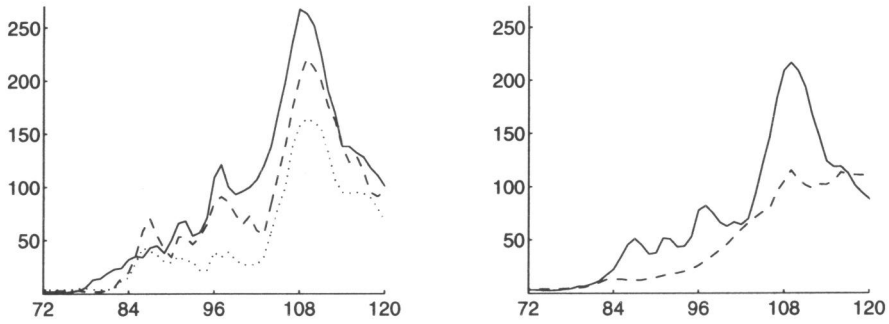


Figure 7.6: SO_2 measurements in $\mu g\, m^{-3}$ for 30-11 and 1-12 1989. Left: station 1 (dotted), 2 (dashed) and 3 (solid); right: averaged measured concentrations over the 3 stations (solid) and modeled concentration in the coarse grid cell using 4 grid levels (dashed). The horizontal axis represents time in hours, 30-11-1989 0:00 GMT = 72.

predicts the peak at the right time, though the modeled peak value is lower than actually observed. After reaching the peak values, the observed concentrations decrease rapidly, whereas the modeled concentrations only slowly decrease. The reason for this is not clear. Wind directions may have changed very quickly. The model applies time interpolation between two 6-hour wind fields and wind variations in the model are therefore always smooth. Another explanation could be the occurrence of precipitation. According to the available precipitation fields derived from synoptic measurements, there was no rain in the area of interest in the selected period. If there has been significant local rainfall, that has not been resolved by the measurements, upwind of (or at) the stations, efficient wash-out would have taken place causing a drop in concentrations as observed, which then would have become visible in the model results, since wet deposition is included in the model. Another possible explanation is a change in mixing height. If the mixing height in-

number	location	θ	ϕ
1	Sappemeer	53.14	6.80
2	Hoogersmilde	52.90	6.40
3	Wijerswold	52.66	6.81
4	Kloosterburen	53.40	6.41
5	Cornjum	53.24	5.61

Table 7.2: Some measurements stations and their coordinates in degrees

creases due to a change in weather conditions, the pollutants will be diluted and their concentrations will decrease. The model, however, will not follow such a change in mixing height, since the mixing height has a predescribed profile and is taken constant in space.

On higher grid levels, the three stations are located in different grid cells. Figure 7.7 contains the modeled concentrations plots for station 1 and 3 for two different grid levels. The prediction for station 1 shows no visible improvement, whereas for station 3 shows less good agreement when using more grid levels. A similar result is obtained for station 2 (not plotted). A

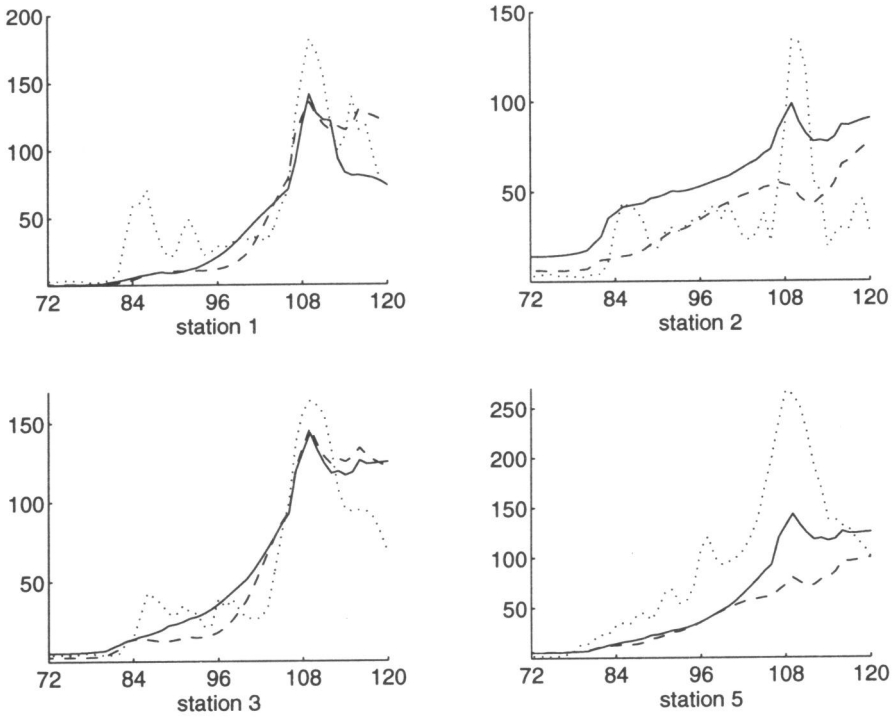


Figure 7.7: Observed (dotted) and modeled SO_2 concentrations in the stations 1,2,3 and 5 using 1 grid level (solid) and 4 grid levels (dashed) in $\mu g\, m^{-3}$ for 30-11 and 1-12 1989

possible explanation is the presence of a local source upwind from station 3, that has not been resolved in the emission inventory of the model. In that case the model can be expected to predict lower values when refining the grid. Another explanation could be the fact that station 3 lies relatively close to the concentration gradient that is present above the Netherlands, as can be seen from the concentration plots in the Figures 7.2-7.5. If the wind fields used in the model are slightly more in northern direction than was actually

the case, large differences like we encounter now may readily occur. Recall that the wind fields have to be obtained by time and spatial interpolation from other wind fields in a different coordinate system, and then have to be made divergence free. The latter process does not only affect the wind speeds, but also (slightly) changes the wind directions. In our experiments, we observed that the wind speeds at the cell centers (of the base grid) are changed by about 10% on average.

Two other station were considered for comparing observations with model predictions. It concerns the stations 4 and 5, also specified in Table 7.2. The observed and modeled concentration profiles can also be found in Figure 7.7. Again we see that the peak values seem to be modeled at the right time. For station 4 the grid refinement does not result into any improvement of the modeled concentration profile. For station 5 however the result of grid refinement is even more disappointing: instead of a closer resemblance with the observed concentrations, the prediction becomes worse, similar as was seen with the modeled concentrations for station 3. The same possible explanations for the differences between modeled and observed concentrations are valid as before.

7.2 The July 1989 episode

For summer smog, no comparison can be made between EUROS and CWIROS. It would make no sense to let EUROS perform ozone calculations as the modeled species in EUROS do not even include the VOCs which are essential in ozone formation. In this section, we present results of model runs for a smog episode from July 1989. The selected period is from 19 July until 24 July, a time interval of 144 hours.

An interesting numerical observation has been made during the experiments. The chemistry has to be solved such that the NO_x balance in each single cell is not disturbed (too much). If one does not take care of this, very different model results are obtained. As the nonlinear system in the chemical kinetics equations, arising from the BDF2 formula, is solved by Gauss-Seidel iteration, see [65, 70], this observation provides a possible stopping criterion for the iteration. In some cells probably the NO_x balance is not disturbed much and two iterations suffice, whereas in other cells 4 or 5 iterations are necessary. In the present experiments the number of iterations was taken equal to 5 in all grid cells. By specifying a stopping criterion, the algorithm may be made more efficient.

7.2.1 Grid refinement

Apart from runs on the base grid, runs with 2, 3 and 4 grid levels have been performed. Table 7.3 gives some information about the number of cells used by the algorithm for the model computations. A comparison with the corresponding Table 7.1 for the winter episode shows that in the present computations more grid cells have been used, especially on the grid levels 3 and 4.

MAXLEV	TOL	grid level 1	2	3	4	total	uniform	%
1	-	2860	-	-	-	2860	-	-
2	0.5	2860	2232	-	-	5092	11440	44.7
3	0.5	2860	2291	6597	-	11747	45760	25.7
4	0.75	2860	2276	6557	12207	23900	183040	13.1

Table 7.3: Average number of cells for the model runs

Both tables show that the grid refinement becomes more efficient for MAXLEV larger than two. However, on our fastest workstation, the computations for MAXLEV>2 took much more computing time than the allowed 3 or 4 hours. This means that the restriction on the area in which the algorithm is allowed to refine the grid, is necessary and even then we have to further restrict the number of cells in order to meet the restriction on the computation time. Of course it is possible to increase the tolerance on the higher grid levels, but for the present experiments this has not been done in order to be sure of the quality of the fine grid solutions.

Figure 7.8 shows the computed concentration distributions over Europe according to a coarse and fine grid computation. Figure 7.8 clearly shows

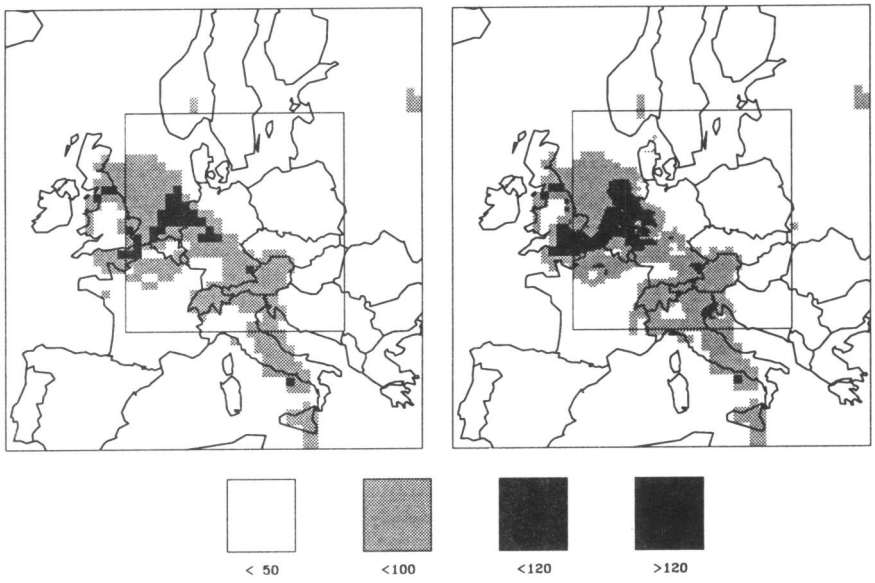


Figure 7.8: Computed O_3 distribution in $\mu g\ m^{-3}$ over Europe for 24-7-1989, 14:00 MET. For the coarse grid (l) and for 4 levels (r).

that grid refinement results into higher ozone concentrations in Europe, especially in the Netherlands and in Great Britain. From the comparisons in

number	location	θ	ϕ
6	Braakman	51.30	3.75
7	Wijnandsrade	50.90	5.88
8	Houtakker	51.52	5.15
9	Hellendoorn	52.39	6.40

Table 7.4: Some measurements stations and their coordinates in degrees

Section 7.2.2 we will see that this means a better agreement with observed concentrations.

7.2.2 Comparison with Dutch measurements

While the winter smog episode was mainly restricted to the North-Eastern part of the Netherlands, in the present summer episode increased ozone concentrations are observed in the whole country. The highest concentrations occur in the South. Therefore we took measurements from three stations in the South together with one station in the North for comparisons with model calculations. The stations are listed in Table 7.4. Figure 7.9 shows the concentrations in the stations 6-9. Note that the measurements series for station 6 is incomplete, unfortunately. Figure 7.9 clearly shows that grid refinement improves the model predictions. Especially the peak values on the last day of the simulations are much better represented in the 4-level computation than in the coarse grid computation. We also see that the observed high values on the third day of the simulation are not predicted by the model and that grid refinement does not show improvement for that particular day. The nightly minimum values for station 9 (the Northern station) seem to be systematically too high. This gives rise to some questions concerning some modeling aspects and not concerning the grid refinement. In general we consider the model results in this comparison to be quite good.

7.2.3 Timings

On the fastest workstation available at CWI, the model runs using up to three grid levels could be done within four hours. Table 7.3 shows that the total (average) number of cells doubles when increasing MAXLEV by one. Hence, roughly speaking, the total CPU time also doubles. The run with MAXLEV=4 took about nine hours of computation time, the run with MAXLEV=3 only four. For on-line application of the model MAXLEV should not be taken larger than three.

7.3 Summary and conclusions

From the experiments described in this chapter the following conclusions are drawn:

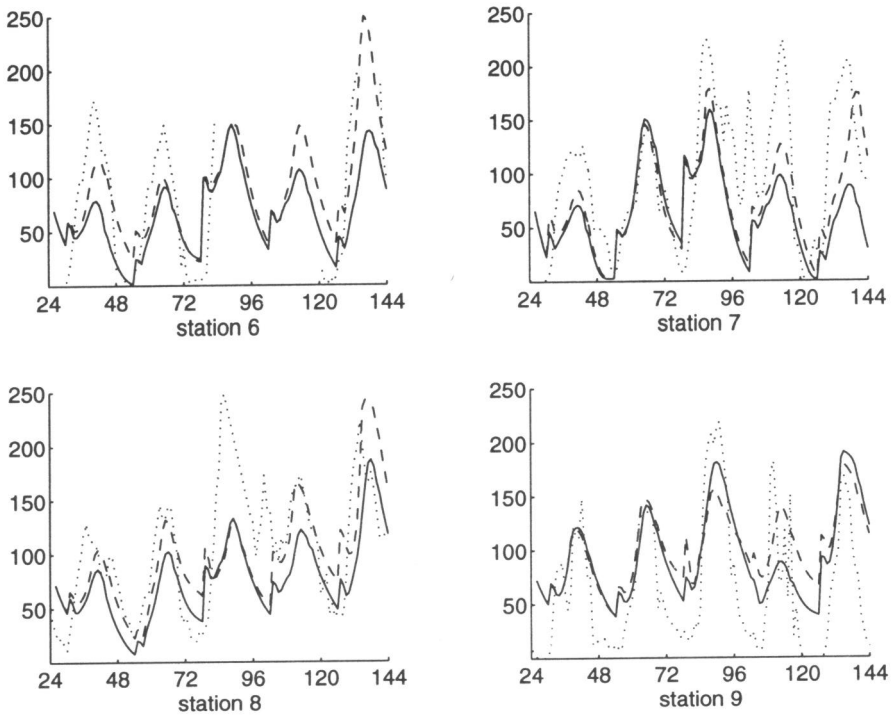


Figure 7.9: Observed (dotted) and modeled concentrations using 1 grid level (solid) and 4 grid levels (dashed) $\mu g\, m^{-3}$

- CWIROS is in accordance with EUROS in a qualitative way. Differences are observed but seem to be caused by different (meteorological) input and not due to (errors in) the computational process.

- The time behavior of the model seems in order. The experiments in Section 7.1 show that the model is able to predict concentration peaks at the right time. This indicates that the emission inventory as well as the advection and emission/deposition routines in the model are correct.

- The grid refinement technique works properly, but a better agreement with observed concentrations has not been obtained by applying this technique in the winter smog episode from November/December 1989. In the July 1989 summer smog episode, the refinement does produce concentration profiles that are in significantly better agreement with measurements.

- Restriction on the number of grid cells of the fine grids may be necessary for summer smog computations due to operational constraints. The

present approach of restricting the area in which the algorithm may refine the grid is probably not sufficient.

- The model seems to be sensitive for meteorological input. Because of uncertainty about the wind fields (and how they are used in the model) combined with uncertainty in other meteorological parameters, the comparisons of modeled concentrations with observed concentrations in Section 7.1 are only indicative. The same uncertainty holds for the summer smog episode, see Section 7.2, even though in that case the model computations are in good agreement with the measurements.

- It has been shown that grid refinement did not result in a (significant) better agreement between observed and modeled concentrations in all cases. A reason for this might be that the improved spatial resolution has not been used in an optimal way. More attention should be paid to bringing the spatial resolution in the description of other (atmospheric) processes in line with the spatial resolution of the fine grids. For example, by using emission inventories with the same resolution as the finest grid (7.5×7.5 km.).

Chapter 8

Summary and conclusions

The project EUSMOG, of which this thesis is one of the scientific results, had a twofold purpose.

The first purpose was the extension of the existing winter smog model EUROS to a summer smog model. This summer smog model got the preliminary name CWIROS, which has, however, not been changed any more during the course of the project. The modeling aspects of this extension were the responsibility of the Dutch National Institute of Public Health and Environmental Protection (RIVM). The RIVM was also the sponsor of the project EUSMOG. Implementation was CWI's responsibility. Most important aspects of the model extension consist of the increased number of modeled species (from 5 to 15) and the much more complex chemical mechanism. In addition, new emission data were used, that also contain point source information. The consequence of these adjustments is that the computation time for one model run drastically increases. In view of the on-line application, the total computation time on a workstation needs to be restricted to 3 or 4 hours.

This lead to the second goal: the development of fast and efficient numerical methods for application in the smog model, as well as the implementation of a local grid refinement technique. Because operator splitting is applied, it is possible to choose a suitable numerical technique for each physical process separately. In the research, attention is paid to

- local uniform grid refinement,
- numerical methods for advection and
- fast and efficient solvers for the chemical equations.

The latter is of great importance, because standard use of stiff ODE solvers would lead to an unproportional amount of computation time for the chemical equations. Because of the restriction to the total computation time, it is impossible to apply standard solvers. However, owing to the relatively low accuracy requirement it is possible to use special purpose solvers. These solvers can be more efficient than standard solvers for relatively low accuracies.

The results and conclusions of the research described in this thesis, can be summarized as follows:

- The development of a compact Eulerian summer smog model has proved to be possible. Its numerical implementation can be run on a worksta-

tion and a model run can be performed within the allowed computation time.

- The grid refinement technique has to be based upon the finite volume approach. This prevents inconsistencies when dealing with (point) sources. The technique presented in this work has been derived from the one by Trompert & Verwer [63, 64]. Since the latter technique is based upon the grid point approach, it has been adapted for finite volumes. Because of the specific application, the datastructure has been extended with a few pointers. Also, the procedures for interpolation of values from a coarse to a fine grid and vice versa have been made mass conserving.

- The method of lines approach has been chosen for the treatment of the advection. In the model, the limited third order $\kappa = \frac{1}{3}$ space discretization with a third order Runge-Kutta time integrator has been implemented. Apart from theoretical investigations on this space discretization and various time integration methods, a comparison is made between the method chosen and other methods. We have investigated whether flux corrected transport could be an alternative for making solutions monotone in case the space discretization is not limited. The (unlimited) third order and fourth order central discretization were considered. However, it turned out that flux corrected transport is computationally expensive, although the accuracies are comparable. Moreover, implementation of flux corrected transport on irregular, refined grids would be even more expensive.

- For the numerical treatment of the chemical equations, a variant on the method TWOSTEP has been chosen. The latter is based on the BDF2 method. The system of nonlinear equations is not approximated in the usual way, by application of Newton's method, but a number of Gauss-Seidel iterations is used instead. For gas-phase atmospheric chemical systems this appears to work well. The iterative process can be improved considerably by means of lumping. For the chemical model, as described in this thesis, lumping of NO_2 and NO into NO_x and NO_2 and O_3 into O_x turns out to be very effective. Instead of TWOSTEP, the variant 3STEP, based on the BDF3 method, is recommended with fixed step sizes of 450 seconds.

  From box experiments with various special purpose solvers, it can be concluded that TWOSTEP and 3STEP are much more efficient than methods based on the QSSA approach. The latter class of methods is widely used in atmospheric models. Also a comparison is made with the state of the art solver VODE, provided with sparse matrix routines to economize on the linear algebra. For relatively small chemical systems, TWOSTEP and 3STEP appeared to be more efficient in the accuracy range of interest. For larger systems (or larger accuracy) VODE often

turns out to be more efficient, provided that the linear algebra has been optimized.

- Compared to measurements, the model results are reasonable or good, in particular for the summer smog episode considered. In most cases, application of grid refinement results into a better agreement between computed and measured concentrations. However, also situations occur in which model computations do not lead to good agreement with measurements. In some cases, application of grid refinement leads to less good agreement between modeled and measured concentrations. In particular, this is the case for the winter smog episode considered. The reason for that can be of a various nature. The modeling of some of the physical processes need to be (re)examined critically.

Bibliography

1. D.C. Arney and J.E. Flaherty. An adaptive local mesh refinement method for time-dependent partial differential equations. *Appl. Numer. Math.*, 5: 257 – 274, 1989.
2. O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, Cambridge, 1994.
3. A.C.M. Beljaars and A.A.M. Holtslag. A software library for the calculations of surface fluxes over land and sea. *Environmental Software*, 5, 1990.
4. M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comp. Phys.*, 82: 64 – 84, 1989.
5. M.J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Phys.*, 53: 484 – 512, 1984.
6. J.G. Blom, R.A. Trompert, and J.G. Verwer. VLUGR2: A Vectorizable Adaptive Grid solver For PDEs in 2D. Report NM–R9403, CWI, Amsterdam (to appear in the June 1996 issue of ACM Trans. Math. Softw.), 1993.
7. J.G. Blom and J.G. Verwer. VLUGR2: A Vectorized Local Uniform Grid Refinement Code for PDEs in 2D. Report NM–R9306, CWI, Amsterdam, 1993.
8. C. Bolley and M. Crouzeix. Conservation de la positivité lors de la discrétisation des problèmes d'évolution paraboliques. *R.A.I.R.O Analyse Numérique*, 12: 237 – 245, 1978.
9. D.L. Book, J.P. Boris, and K. Hain. Flux corrected transport II: Generalizations of the method. *J. Comp. Phys.*, 18: 248 – 283, 1975.
10. J.P. Boris and D.L. Book. Flux corrected transport I: SHASTA, a fluid transport algorithm that works. *J. Comp. Phys.*, 11: 38 – 69, 1973.
11. J.P. Boris and D.L. Book. Flux corrected transport III: Minimal-error FCT algorithms. *J. Comp. Phys.*, 20: 397 – 431, 1976.
12. P.N. Brown, G.D. Byrne, and A.C. Hindmarsh. VODE: A variable coefficient ODE solver. *SIAM J. Sci. Stat. Comput.*, 10: 1038 – 1051, 1989.
13. H.A.R. de Bruin and A.A.M. Holtslag. A Simple Parameterization of the Surface Fluxes of Sensible and Latent Heat During Daytime Compared with the Penman-Monteith Concept. *Journal of Applied Meteorology*, 21: 1610 – 1621, 1982.
14. J. Burn. Smog. *European Bulletin on Environment and Health*, 1, No.2: 3 – 6, 1992.
15. K. Dekker and J.G. Verwer. *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations.* North-Holland, Amsterdam, The Netherlands, 1984.

16. J.J. Dongarra and E. Grosse. Distribution of software via electronic mail. *Commun. ACM*, 30: 403 – 407, 1987. (netlib@research.att.com).

17. H. van Dop and B.J. de Haan. Mesoscale air pollution dispersion modelling. *Atm. Environment*, 17: 1449 – 1456, 1983.

18. R.M. Endlich. An iterative method for altering the kinematic properties of wind fields. *Journal of Applied Meteorology*, 6: 837 – 844, 1967.

19. F.A.A.M. de Leeuw et al. Een Lagrangiaans lange-afstand transport model met niet-lineaire atmosferische chemie: MPA-model. Technical Report RIVM report 228471004; TNO report R87/344, RIVM Bilthoven (NL); TNO Delft (NL), 1988. *In Dutch.*

20. R. D. Grigorieff. Stability of multistep-methods on variable grids. *Numer. Math.*, 42: 359 – 377, 1983.

21. W.D. Gropp. A test of moving mesh refinement for 2D-scalar hyperbolic problems. *SIAM J. Sci. Statist. Comput.*, 1: 191 – 197, 1980.

22. W.D. Gropp. Local uniform mesh refinement on vector and parallel processors. In P. Deuflhard and B. Engquist, editors, *Large–Scale Scientific Computing*, Birkhäuser Series Progress in Scientific Computing 7, pages 349 – 367. Birkhäuser, Basel, 1987.

23. W.D. Gropp. Local uniform mesh refinement with moving grids. *SIAM J. Sci. Statist. Comput.*, 8: 292 – 304, 1987.

24. E. Hairer and G. Wanner. *Solving ordinary differential equations II – Stiff and differential algebraic problems.* Springer Series in Computational Mathematics 8. Springer–Verlag, Berlin, 1991.

25. O. Hertel, R. Berkowicz, and J. Christensen. Test of two numerical schemes for use in atmospheric transport-chemistry models. *Atm. Environment*, 16: 2591 – 2611, 1993.

26. E. Hesstvedt, Ø. Hov, and I.S.A. Isaksen. Quasi steady-state approximation in air pollution modeling: Comparison of two numerical schemes for oxidant prediction. *Int. J. Chem. Kinet.*, 10: 971 – 994, 1978.

27. Ø. Hov, I.S.A. Isaksen, and E. Hesstvedt. A numerical method to predict secondary air pollutants with an application on oxidant generation in an urban atmosphere. In *WMP Symposium on Boundary Layer Physics Applied to Specific Problems of Air Pollution*, pages 219 – 226, Geneva, Switzerland, 1978. WMO Publication No. 510.

28. W. Hundsdorfer, B. Koren, M. van Loon, and J.G. Verwer. A Positive Finite-Difference Advection Scheme Applied on Locally Refined Grids. Report NM–R9309, CWI, Amsterdam, 1993.

29. W. Hundsdorfer, B. Koren, M. van Loon, and J.G. Verwer. A Positive Finite-Difference Advection Scheme. *J. Comp. Phys.*, 117: 35 – 46, 1995. Revision of CWI report NM–R9309.

30. W. Hundsdorfer and E.J. Spee. An efficient horizontal advection scheme for the modeling of global transport of constituents. *Mon. Wea, Rev*, 123: 3554 – 3564, 1995. Revision of CWI Report NM–R9416.

31. W. Hundsdorfer and R.A. Trompert. Method of lines and direct discretization: a comparison for linear advection. *App. Num. Math.*, 13: 469 – 490, 1994.

32. M.Z. Jacobson and R.P. Turco. SMVGEAR: A Sparse-Matrix, Vectorized Gear

Code for Atmospheric Models. *Atm. Environment*, 28: 273 – 284, 1994.

33. L. Jay, A. Sandu, F. Potra, and G. Carmichael. Improved QSSA methods for Atmospheric chemistry integration. Reports on Computational Mathematics no. 67, The Univ. of Iowa, Iowa City, IA 52242, February 1995.
34. B. Koren. A robust upwind discretization method for advection, diffusion and source terms. In C.B. Vreugdenhil and B. Koren, editors, *Notes on Numerical Fluid Mechanics, Chapter 5*, volume 45. Vieweg, Braunschweig, 1993.
35. J.F.B.M. Kraaijevanger. Absolute monotonicity of polynomials occurring in the numerical solution of initial value probems. *Numer. Math.*, 48: 303 – 322, 1986.
36. B. van Leer. Upwind–difference methods for aerodynamic problems governed by the Euler equations. In B.E. Engquist, S. Osher, and R.C.J. Somerville, editors, *Large–scale computations in fluid mechanics*, pages 327 – 336. AMS Series, American Mathematical Society, Providence, RI, 1985.
37. F.A.A.M. de Leeuw. Een een-dimensionaal diffusiemodel: modelconcept en enkele toepassingen. Technical Report RIVM report 228603001, RIVM Bilthoven (The Netherlands), 1987. *In Dutch.*
38. F.A.A.M. de Leeuw. Private communication. 1994.
39. F.A.A.M. de Leeuw, H. Kesseboom, and N.D. van Egmond. Numerieke verspreidingsmodellen voor de interpretatie van meetresultaten van het Nationaal Meetnet voor Luchtverontreiniging; ontwikkeling 1982-1985. Report 842017002, National Institute of Public Health and Environmental Protection (RIVM), Bilthoven, The Netherlands, 1985. *In Dutch.*
40. R.J. LeVeque. *Numerical methods for conservation laws.* Lecture Notes in Mathematics ETH Zűrich. Birkhäuser Verlag, Basel, 1992.
41. M. van Loon. Testing interpolation and filtering techniques in connection with a semi-lagrangian method. *Atm. Environment*, 27A: 2351 – 2364, 1993.
42. M. van Loon. Numerical smog prediction I: The physical and chemical model. Report NM–R9411, CWI, Amsterdam, 1994.
43. M. van Loon. Fast and efficient solution methods for ozone chemistry. In H. Power, N. Moussiopoulos, and C.A. Brebbia, editors, *Air pollution III Volume 1: Theory and Simulation*, pages 335 – 342, Computational Mechanics Publications, Southampton Boston, 1995.
44. P.C. Manins. Partial penetration of an elevated inversion layer by chimney plumes. *Atm. Environment*, 13: 733 – 741, 1979.
45. J.E. McDonald. Saturation vapor pressure over supercooled water. *J. Geophys. Res.*, 70: 1552 – 1554, 1965.
46. 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: 1 – 42, 1982.
47. H. J. van Rheineck Leyssius and F.A.A.M. de Leeuw. Prognose van luchtkwaliteit: signalering van fotochemische smogepisoden. Report 222106001, National Institute of Public Health and Environmental Protection (RIVM), Bilthoven, The Netherlands, 1990. *In Dutch.*
48. H. J. van Rheineck Leyssius and F.A.A.M. de Leeuw. Prognose van luchtkwaliteit: signalering van wintersmogepisoden. Report 222106002, National

Institute of Public Health and Environmental Protection (RIVM), Bilthoven, The Netherlands, 1991. *In Dutch.*

49. H.J. van Rheineck Leyssius, F.A.A.M. de Leeuw, and Bert H. Kesseboom. A regional scale model for the calculation of episodic concentrations and depositions of acidifying components. *Water, Air and Soil Pollution*, 51: 327 – 344, 1990.
50. J.S. Rosenbaum. Conservation Properties of Numerical Integration Methods for Systems of Ordinary Differential Equations. *J. Comp. Phys.*, 20: 259 – 267, 1976.
51. A. Sandu. Private Communication. 1996.
52. A. Sandu, J.G. Verwer, M. van Loon, G.R. Carmichael, F.A. Potra, D. Dabdub, and J.H. Seinfeld. Benchmarking Stiff ODE Solvers for Atmospheric Chemistry Problems: Implicit versus Explicit. *submitted to Atm. Environment*, 1996.
53. A.H. Sherman and A.C. Hindmarsh. GEARS: a package for the solution of sparse, stiff ordinary differential equations. Technical Report UCRL-84102, 1080, Lawrence Livermore Laboratory.
54. C.-W. Shu and S. Osher. Efficient implementation of essentially non–oscillatory shock–capturing schemes. *J. Comp. Phys.*, 77: 439 – 471, 1988.
55. C.-W. Shu and S. Osher. Efficient implementation of essentially non–oscillatory shock–capturing schemes II. *J. Comp. Phys.*, 83: 32 – 78, 1989.
56. D. Simpson. Biogenic VOC in Europe. Part II: implications for ozone control strategies. *submitted to Atm. Environment*, 1994.
57. D. Simpson, Y. Andersson-Skőld, and M.E. Jenkin. Updating the chemical scheme for the EMEP MSC-W oxidant model: current status. Technical Report note 2/93, EMEP MSC-West, Oslo, Norway, 1993.
58. J.R. Stedman and M.L. Williams. A trajectory model of the relationship between ozone and precursor emissions. *Atm. Environment*, 26A: 1271 – 1281, 1992.
59. G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5: 506 – 517, 1968.
60. P. Sun, D.P. Chock, and S.L. Winkler. An Implicit-Explicit Hybrid Solver for a System of Stiff Kinetics Equations. *J. Comp. Phys.*, 115: 515 – 523, 1994.
61. P.K. Sweby. High resolution schemes using flux–limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 21: 995 – 1011, 1984.
62. H. The. An accurate and fast numerical algorithm for solving sets of stiff differential equations in atmospheric chemistry suitable for large time steps. Technical report, Institute for Environmental Protection and Hygiene (RIVM), Bilthoven, The Netherlands, 1994.
63. R.A. Trompert and J.G. Verwer. A static–regridding method for two–dimensional parabolic partial differential equations. *Appl. Numer. Math.*, 8: 65 – 90, 1991.
64. R.A. Trompert and J.G. Verwer. Analysis of the implicit Euler local uniform grid refinement method. *SIAM J. Sci. Comput.*, 14: 259 – 278, 1993.
65. J.G. Verwer. Gauss-Seidel iteration for stiff ODEs from chemical kinetics. *SIAM J. Sci. Comput.*, 15: 1243 – 1250, 1994.

66. J.G. Verwer, J.G. Blom, and W.H. Hundsdorfer. An implicit-explicit approach for atmospheric transport-chemistry problems. *Appl. Numer. Math.*, 20: 191 – 209, 1996.

67. J.G. Verwer, J.G. Blom, M. van Loon, and E.J. Spee. A Comparison of Stiff ODE Solvers for Atmospheric Chemistry Problems. *Atm. Environment*, 30: 49 – 58, 1996.

68. J.G. Verwer, W.H. Hundsdorfer, and J.G. Blom. Convergence of Jacobi and Gauss-Seidel Iteration for Stiff Atmospheric Differential Equations. Internal note, CWI, Amsterdam, 1995. (unpublished).

69. J.G. Verwer and M. van Loon. An evaluation of explicit pseudo-steady-state approximation schemes for stiff ODEs from chemical kinetics. *J. Comp. Phys.*, 113: 347 –352, 1993.

70. J.G. Verwer and D. Simpson. Explicit methods for stiff ODEs from atmospheric chemistry. *Appl. Numer. Math.*, 18: 413 –430, 1995.

71. D.L. Williamson. Review of numerical approaches for modeling global transport. In H. van Dop and G. Kallos, editors, *Air pollution Modeling and its applications IX*, pages 377 – 394, Plenum press, New York, 1992.

72. D.L. Williamson and P.J. Rasch. Two-dimensional semi-Lagrangian transport with shape-preserving interpolation. *Mon. Weath. Rev.*, 117: 102 – 129, 1989.

73. K. Wirtz, C. Roehl, G.D. Hayman, and M.E. Jenkin. LACTOZ evaluation of the EMEP MSC-W photo-oxidant model. Technical report, EUROTRAC ISS, Garmisch Partenkirchen, Germany, 1994.

74. S.T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comp. Phys.*, 31: 335 – 362, 1979.

75. P. Zanetti. *Air Pollution Modeling, theories, computational methods and available software.* Computational Mechanics Publications Southampton Boston. Van Nostrand Reinhold, New York, 1990.

Samenvatting

Het project EUSMOG, waarvan dit proefschrift één van de wetenschappelijke resultaten is, had een tweeledig doel.

Het eerste doel is het uitbreiden van het bestaande wintersmog model EUROS naar een zomersmog model. Dit zomersmog model heeft in eerste instantie de naam CWIROS gekregen. Gedurende het project is deze naam echter niet meer gewijzigd. Voor deze modeluitbreiding was het Rijksinstituut voor Volksgezondheid en Mileuhygiëne (RIVM), tevens opdrachtgever van het project EUSMOG, de eerstverantwoordelijke. De implementatie kwam echter voor rekening van het CWI. Belangrijkste aspecten van de bedoelde modeluitbreiding waren het grotere aantal gemodelleerde chemische componenten (van 5 naar 15) en een veel ingewikkelder chemisch reactiemechanisme. Tevens werden nieuwe emissiebestanden gebruikt die ook puntbroninformatie bevatten. Het gevolg van deze aanpassingen was dat de benodigde rekentijd op een workstation drastisch zou toenemen. Echter, de rekentijd diende beperkt te blijven tot 3 à 4 uur vanwege de praktische toepassing.

Dit leidde tot het tweede doel: het ontwikkelen van snelle en efficiënte numeriek-wiskundige methoden voor toepassing in het smog model, alsmede de implementatie van een locale roosterverfijningstechniek. Omdat in het model operator splitting wordt toegepast, is het mogelijk voor ieder fysisch proces afzonderlijk een geschikte numerieke techniek te kiezen. In het onderzoek is met name aandacht gegeven aan numerieke methoden voor advectie, daarbij rekening houdend met de gekozen gridverfijningstechniek, en snelle en efficiënte solvers voor chemische vergelijkingen. Dit laatste is van groot belang omdat bij standaard gebruik van stijve ODE solvers een onevenredig groot deel van de rekentijd zou gaan zitten in het oplossen van chemische vergelijkingen. Gezien de gestelde restrictie aan de totale rekentijd van het model, is het onmogelijk standaard methoden toe te passen. Vanwege de relatief lage nauwkeurigheidseis is het mogelijk special-purpose solvers te gebruiken. Deze kunnen efficiënter zijn dan standaard solvers voor relatief lage nauwkeurigheden.

De resultaten en conclusies van het onderzoek gerapporteerd in dit proefschrift kunnen als volgt worden samengevat:

- Het is mogelijk gebleken een compact Euleriaans zomersmog model te ontwikkelen. De numerieke implementatie ervan kan draaien op een workstation en een modelrun kan worden uitgevoerd binnen de gestelde rekentijd.

- De gridverfijningstechniek dient gebaseerd te zijn op de eindige vo-

lume aanpak. Dit voorkomt inconsistenties bij de behandeling van (punt)bronnen. De gekozen aanpak is gebaseerd op de methode ontwikkeld door Trompert & Verwer [63, 64]. Deze aanpak is echter gebaseerd op een grid punt interpretatie en is daarom aangepast voor de eindige volume benadering. Vanwege de toepassing is tevens de datastructuur uitgebreid met een aantal pointers. Tevens zijn de procedures voor het interpoleren van oplossingen van een grof naar een fijn grid en omgekeerd massabehoudend gemaakt.

- Voor de behandeling van de advectie is gekozen voor de methode der lijnen benadering. In het model is de gelimite $\kappa = \frac{1}{3}$ plaatsdiscretisatie met een derde orde Runge-Kutta tijdsintegrator geïmplementeerd. Naast theoretisch onderzoek naar deze discretisatie en verschillende tijdsintegratie methoden, is een vergelijking gemaakt tussen de gekozen methode en andere methodes. Daarbij is onderzocht of flux corrected transport een alternatief kan zijn voor het monotoon maken van de oplossing in geval de plaatsdiscretisatie niet gelimit wordt. Naast de niet gelimite derde orde plaatsdiscretisatie is ook de vierde orde centrale discretisatie beschouwd. Het bleek echter dat flux corrected transport weliswaar vergelijkbaar nauwkeurig is, maar ook duur. Implementatie ervan op verfijnde roosters zou bovendien nog duurder uitpakken dan op een regelmatig rooster.

- Voor de numerieke behandeling van de chemische vergelijkingen is gekozen voor een variant op de methode TWOSTEP. Deze laatste methode is gebaseerd op de BDF2 methode. Het stelsel niet-lineaire vergelijkingen wordt echter niet met behulp van de methode van Newton opgelost (benaderd), maar door een aantal Gauss-Seidel iteraties. Dit blijkt goed te werken voor gas-fase atmosferische chemische systemen. Het iteratieve proces kan worden verbeterd door middel van lumping. Voor het chemische model, zoals beschreven in dit proefschrift, blijkt lumping van NO_2 en NO in NO_x en van NO_2 en O_3 in O_x zeer effectief. In plaats van TWOSTEP wordt de variant 3STEP, gebaseerd op de BDF3 methode, aanbevolen met gebruik van vaste tijdstappen van 450 seconden.

  Uit box experimenten met verschillende special-purpose solvers bleek dat TWOSTEP en 3STEP veel efficiënter zijn dan methoden gebaseerd op de QSSA benadering. Deze laatste klasse van methoden wordt veel gebruikt in atmosferische modellen. Tevens is een vergelijking gemaakt met de state-of-the-art solver VODE, voorzien van ijle matrix routines om de lineaire algebra efficiënter te maken. Voor relatief kleine chemische modellen bleken TWOSTEP en 3STEP efficiënter in het gewenste nauwkeurigheidsgebied. Voor grotere systemen (dan wel voor hogere nauwkeurigheid) blijkt VODE vaak efficiënter te kunnen zijn, mits de lineaire algebra geoptimaliseerd is.

- In vergelijking met meetresultaten zijn de modeluitkomsten redelijk tot goed te noemen, met name voor de beschouwde zomersmog epi-

sode. Toepassing van gridverfijning resulteert in een aantal gevallen tot een betere overeenkomst tussen berekende en gemeten concentraties. Er zijn echter ook situaties waarin de modelberekeningen geen goede overeenkomst vertonen met metingen. In een enkel geval leidt gridverfijning tot een minder goede overeenstemming tussen gemodelleerde en gemeten concentraties. Dit is met name het geval voor de beschouwde winterepisode. De oorzaken hiervan kunnen van diverse aard zijn. Er zal nog kritisch gekeken moeten worden naar de modellering van verschillende processen.

De indeling van dit proefschrift is als volgt:

Na het inleidende Hoofdstuk 1 wordt in Hoofdstuk 2 het fysische en chemische model beschreven.

Hoofdstuk 3 beschrijft de gridverfijningstechniek. Tevens wordt in dit hoofdstuk een numerieke illustratie gegeven.

Hoofdstuk 4 behandelt numerieke advectie schema's volgens de eindige volume benadering. Uitgangspunt daarbij is massabehoud en het voorkomen van under- en overshoot. De nadruk in dit hoofdstuk ligt op de gelimite $\kappa = \frac{1}{3}$ plaatsdiscretisatie gecombineerd met Runge-Kutta tijdsintegratie. Deze combinatie levert een massabehoudend schema op. Condities voor de tijdstap worden afgeleid waarbij de numerieke oplossing vrij blijft van under- en overshoot. Daarnaast wordt ook Flux Corrected Transport (FCT) besproken als mogelijkheid om het laatste te bereiken. Het hoofdstuk besluit met numerieke experimenten met een aantal geselecteerde advectie schema's.

Hoofdstuk 5 beschrijft special-purpose solvers voor het oplossen van chemische vergelijkingen. Deze methoden zijn vaak niet massabehoudend of zijn dat alleen wanneer men een iteratief proces laat convergeren. In dit hoofdstuk wordt daarom ook aandacht gegeven aan technieken om de massabalans te verbeteren. Voor TWOSTEP en QSSA methoden wordt enige analyse gegeven voor lineaire chemie, zowel met betrekking tot convergentie als massabehoud.

Hoofdstuk 6 bevat een numerieke vergelijking tussen verschillende chemische solvers door middel van een box model test. Deze test is zo opgezet dat zoveel mogelijk recht wordt gedaan aan de praktische toepassing van de solvers in het volledige model.

Hoofdstuk 7 laat resultaten zien van modelberekeningen, gedaan met verschillende roosterniveau's, en legt deze naast waarnemingen. Experimenten voor een wintersmog en een zomersmog periode worden gepresenteerd.

Hoofdstuk 8, tenslotte, geeft een samenvatting van dit proefschrift en bevat tevens de conclusies uit het onderzoek in het kader van het project EUSMOG.

Appendix A

Shifted pole coordinates

A.1 Transforming θ and ϕ

On the globe we define spherical coordinates. They are given by

$$\begin{aligned} x &= r\cos\theta\cos\phi \\ y &= r\cos\theta\sin\phi \\ z &= r\sin\theta \end{aligned} \tag{A.1}$$

where ϕ and θ are the longitude and latitude in radians, respectively. The (constant) radius of the earth is denoted by r. So the equator lies in the plane $z = 0$ and the meridian through Greenwich in the plane $y = 0$. We will describe the coordinate system resulting from a shift of the pole, or rather the equator. This shift can be seen as rotation of the xy-plane around the y-axis under an angle α, $0 \leq \alpha \leq \frac{\pi}{2}$, yielding a new coordinate system $(\bar{x}, \bar{y}, \bar{z})$ which corresponds with $(\bar{\theta}, \bar{\phi})$ similar as in (A.1). It can be seen that

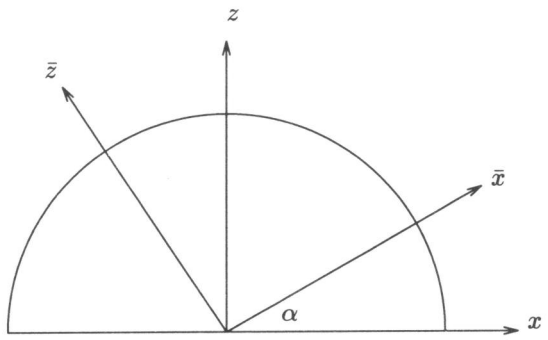


Figure A.1: Schematic representation of the transformation

$$\begin{aligned} x &= \bar{x}\cos\alpha - \bar{z}\sin\alpha \\ y &= \bar{y} \\ z &= \bar{x}\sin\alpha + \bar{z}\cos\alpha \end{aligned} \tag{A.2}$$

Combining (A.1) and (A.2) gives

$$\begin{aligned} \bar{x} &= z\sin\alpha + x\cos\alpha \\ \bar{z} &= z\cos\alpha - x\sin\alpha \end{aligned} \tag{A.3}$$

which, after substitution, results into

$$\begin{aligned} \bar{\theta} &= \arcsin\{\sin\theta\cos\alpha - \cos\theta\cos\phi\sin\alpha\} \\ \bar{\phi} &= \arccos\left\{\frac{\sin\theta\sin\alpha + \cos\theta\cos\phi\cos\alpha}{\cos\bar{\theta}}\right\} \end{aligned} \tag{A.4}$$

It can be seen by substitution of $\phi = 0$ that indeed $\bar{\theta} = \theta - \alpha$ which in return gives $\bar{\phi} = \arccos(1) = 0$. The inverse transformation can easily be made by interchanging $(\bar{\theta}, \bar{\phi})$ and (θ, ϕ) and substituting $-\alpha$ for α in (A.4). Note that for the implementation of (A.4) one needs to make sure that the arguments of the arcsin and arccos are in the interval [-1,1]. It was found that otherwise problems may arise due to rounding errors.

A.2 Transforming u and v

In a similar way we derive the velocities $\bar{u}$ and $\bar{v}$ in $\bar{\theta}$ in $\bar{\phi}$ direction, respectively. Given the wind field (u, v) we decompose the wind field into components w_x, w_y, z_y in x, y and z direction

$$\begin{aligned} w_x &= u\sin\phi - v\sin\theta\cos\phi \\ w_y &= u\cos\phi - v\sin\theta\sin\phi \\ w_z &= v\cos\theta \end{aligned} \tag{A.5}$$

These components are used to obtain the components in $\bar{x}, \bar{y}$ and $\bar{z}$ direction.

$$\begin{aligned} w_{\bar{x}} &= v\cos\theta\sin\alpha + (u\sin\phi - v\sin\theta\cos\phi)\cos\alpha \\ w_{\bar{y}} &= w_y \\ w_{\bar{z}} &= v\cos\theta\cos\alpha - (u\sin\phi - v\sin\theta\cos\phi)\sin\alpha \end{aligned} \tag{A.6}$$

Using $w_{\bar{z}} = \bar{v}\cos\bar{\theta}$ we obtain

$$\bar{v} = \left\{\frac{v\cos\theta\cos\alpha - (u\sin\phi - v\sin\theta\cos\phi)\sin\alpha}{\cos\bar{\theta}}\right\}. \tag{A.7}$$

The corresponding relation for $w_{\bar{y}}$ yields

$$\bar{u} = \left\{\frac{u\cos\phi - v\sin\theta\sin\phi + \bar{v}\sin\bar{\theta}\sin\bar{\phi}}{\cos\bar{\phi}}\right\}. \tag{A.8}$$

Another way to describe the transformation of the wind vector is to consider the rotation of the coordinate axis. For a certain point (ϕ, θ) on the globe, the coordinate axis of the shifted pole coordinates are obtained by rotating the coordinate axis of the usual spherical coordinates over an angle β. This angle β is different for each point. The velocities $\bar{u}$ and $\bar{v}$ are then given by

$$\begin{aligned} \bar{u} &= u\cos\beta - v\sin\beta, \\ \bar{v} &= u\sin\beta + v\cos\beta. \end{aligned}$$

From (A.7) it follows that

$$\sin\beta = -\frac{\sin\phi}{\cos\bar{\theta}}.$$

Appendix B

Time factors for emissions

source category	jan 1	feb 2	mar 3	apr 4	may 5	june 6
1	1.07	1.07	1.07	0.93	0.93	0.93
2	1.55	1.55	1.55	0.45	0.45	0.45
3	1.20	1.20	1.20	0.80	0.80	0.80
4	1.00	1.00	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00	1.00	1.00

source category	juli 7	aug 8	sept 9	oct 10	nov 11	dec 12
1	0.93	0.93	0.93	1.07	1.07	1.07
2	0.45	0.45	0.45	1.55	1.55	1.55
3	0.80	0.80	0.80	1.20	1.20	1.20
4	1.00	1.00	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00	1.00	1.00

Table B.1: γ_m values per source category

source category	mon 1	tue 2	wed 3	thu 4	fri 5	sat 6	sun 7
1	1.07	1.07	1.07	1.07	1.07	0.83	0.83
2	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3	1.06	1.06	1.06	1.06	1.06	0.85	0.85
4	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5	1.26	1.26	1.26	1.26	1.26	0.35	0.35
6	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table B.2: γ_d values per source category

source category	hour					
	1	2	3	4	5	6
1	0.83	0.83	0.83	0.83	0.83	0.83
2	0.50	0.50	0.50	0.50	0.50	0.50
3	0.70	0.70	0.70	0.70	0.70	0.70
4	1.00	1.00	1.00	1.00	1.00	1.00
5	0.10	0.10	0.10	0.10	0.10	0.10
6	0.20	0.20	0.20	0.20	0.20	0.20

source category	hour					
	7	8	9	10	11	12
1	0.83	1.17	1.17	1.17	1.17	1.17
2	0.50	1.50	1.50	1.50	1.50	1.50
3	0.70	1.30	1.30	1.30	1.30	1.30
4	1.00	1.00	1.00	1.00	1.00	1.00
5	0.10	1.90	1.90	1.90	1.90	1.90
6	0.20	1.80	1.80	1.80	1.80	1.80

source category	hour					
	13	14	15	16	17	18
1	1.17	1.17	1.17	1.17	1.17	1.17
2	1.50	1.50	1.50	1.50	1.50	1.50
3	1.30	1.30	1.30	1.30	1.30	1.30
4	1.00	1.00	1.00	1.00	1.00	1.00
5	1.90	1.90	1.90	1.90	1.90	1.90
6	1.80	1.80	1.80	1.80	1.80	1.80

source category	hour					
	19	20	21	22	23	24
1	1.17	0.83	0.83	0.83	0.83	0.83
2	1.50	0.50	0.50	0.50	0.50	0.50
3	1.30	0.70	0.70	0.70	0.70	0.70
4	1.00	1.00	1.00	1.00	1.00	1.00
5	1.90	0.10	0.10	0.10	0.10	0.10
6	1.80	0.20	0.20	0.20	0.20	0.20

Table B.3: γ_h values per source category

Appendix C

Datastructure for grid refinement

C.1 Approach

The way we look at a grid at a certain level is in fact not different from the way we treat a standard rectangular uniform $M \times N$ grid where each cell is identified by its row- and column number. Although the grid cells are numbered from 1 to $M \times N$ instead of using a double index, information about the row- and column number of each cell is available. The row- and column number of a certain fine grid cell are equal to the row- and column number the same cell would have if it was part of a uniform fine $M_{fine} \times N_{fine}$ grid over the whole model domain, where rows and columns are numbered from 0 to $M_{fine} - 1$ and $N_{fine} - 1$, respectively. In Figure C.1 an example is given of a rectangular 6×6 base grid (dashed lines) with one level of refinement, consisting of 40 cells. The bold numbers indicate the cell numbers in the fine grid, the other numbers are coarse grid cell numbers. The idea is to store the solution values row-wise. For each grid level the number of really existing rows is specified (6 for the fine grid in Figure C.1) and indicate for each row where it starts. In addition, for each cell the indices of the cells directly above and below are specified. Also for each grid cell its column number has to be specified. With all this information, discretisations of differential equations on the fine grid can be implemented in quite a simple way. Below, a specification of all (integer) arrays used to describe the grid structure, is given.

C.2 Integer array(s)

The gridstructure for each grid level is contained in one large integer array ISTRUC consisting of the following arrays:

- LROW(0:LROW(0)+1)
  LROW(0): the number of rows in the grid
  LROW(1:NROWS): pointers to the start of each row in the grid
  LROW(NROWS+1): NPTS + 1, the total number of grid cells + 1

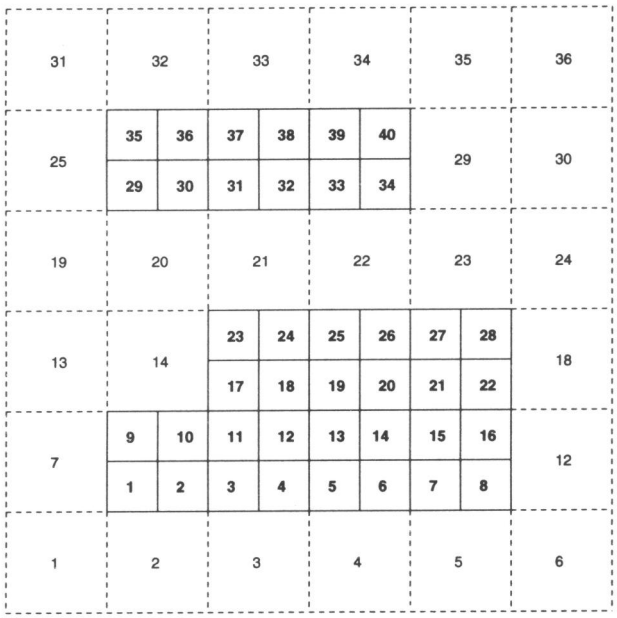


Figure C.1: Example of a two level grid structure

- IROW(NROWS): the row number of each row in a virtual rectangle, corresponding to its θ-coordinate,

- ICOL(NPTS): the column number of a grid cell in the virtual rectangle, corresponding to its ϕ-coordinate

- IPREV(NPTS): pointers to the underlying coarse grid cells; filled with zeros for the base grid,

- LBND(0:LBND(0)+LBND(1)+1)
  LBND(0): NFBPTS, the total number of physical boundary cells in the actual grid level
  LBND(1): NIBPTS, the total number of internal boundary cells in the actual grid level
  LBND(2:NFBPTS+1): pointers to the physical boundary cells in the grid
  LBND(NFBPTS+2:NFBPTS+NIBPTS):
  pointers to the internal boundary cells in the grid,

- LABOVE(0:NPTS): pointers to the node directly above each grid cell, zero otherwise.
  LABOVE(0) is zero, which makes recursive use of LABOVE possible,

- LBELOW(0:NPTS): identical to LABOVE but pointing to the cell directly below,
- INEXT(NPTS): if a cell is refined the corresponding entry of INEXT points to the lower left of the four created fine grid cell on the next level. Otherwise the pointer is set to zero. The pointers are only set when actually creating a next finer grid level, so they are unknown when integrating on the present grid level.

For each of the arrays listed above, we will give (some) values for the fine grid in Figure C.1

- LROW[0] = 6; LROW[1..6] = [1,9,17,23,29,35,41]
- IROW[1..6] = [2,3,4,5,8,9]
- ICOL[1..8] = [2,3,4,5,6,7,8,9]
- IPREV[1] = 8; IPREV[39]=28
- LBND[0,1] = [0,30]; LBND[i+1] = i, $i = 1 \ldots 10$
- LABOVE[0] = 0; LABOVE[i] = $i + 8$, $i = 1 \ldots 8$; LABOVE[9]=0
- LBELOW[0..8] = 0; LBELOW[i] = $i - 8$, $i = 9 \ldots 16$
- INEXT[i] = 0, $i = 1 \ldots 40$
  N.B. for the base grid we have INEXT[7..12] = [0,1,3,4,7,0]

Appendix D

Mass conservation for implicit BDF methods

Consider the nonlinear ODE

$$\dot{y} \;=\; f(y), \quad y \in R^k \tag{D.1}$$

and the general BDF formula

$$y^{n+1} \;=\; Y^n \;+\; \gamma\tau f(y^{n+1}) \tag{D.2}$$

to approximately solve this ODE. Suppose that one or more vectors w exist for which $w^T f(y) = 0$ for all vectors y, where w is a k-vector with constant, nonnegative weights. If $w^T f(y) = 0$, then the exact solution of (D.1) satisfies a mass conservation relation of the form

$$w^T y(t) \;=\; M, \quad M \text{ constant.} \tag{D.3}$$

In Section 5.2.1 it was already shown that the exact solution of the BDF formula (D.2) also satisfies the relation (D.3). We now show, along the lines of Rosenbaum [50], that mass is also preserved if the solution for y^{n+1} of (D.2) is approximated by any finite number of modified Newton iterations. Let $y^{(l)}$ denote the l-th iterate for y^{n+1}. The modified Newton process to obtain the $(l+1)$-th iterate can be written as

$$(I - \gamma\tau J(\tilde{y}))(y^{(l+1)} - y^{(l)}) \;=\; Y^n \;+\; \gamma\tau f(y^{(l)}) - y^{(l)}, \tag{D.4}$$

where J denotes the Jacobian matrix of f. Its argument $\tilde{y}$ is arbitrary and need not be equal to $y^{(0)}$, since implicit solvers usually keep the Jacobian fixed during a number of time steps. Because $w^T f(y) = 0$ for arbitrary y, $w^T J(y)$ is zero as well for arbitrary y. Multiplication of (D.4) by w^T gives

$$w^T(y^{(l+1)} - y^{(l)}) \;=\; w^T(Y^n - y^{(l)}) \;=\; 0,$$

since $w^T Y^n = M$ because Y^n is a linear combination of mass conserving solutions at previous time levels and the sum of the weights is one. Hence, the approximate solution for y^{n+1} of (D.2), obtained by any number of modified Newton iterations, is mass conserving. This is even true if the start vector $y^{(0)}$ is not mass conserving. Multiplication of (D.4) by w^T for $l = 0$ then gives

$$w^T(y^{(1)} - y^{(0)}) = M - w^T y^{(0)},$$

or, equivalently,

$$w^T y^{(1)} = M.$$

In other words, modified Newton iteration is mass conserving for any number of iterations, independent of the start vector. In a similar way, it can be shown that mass is also preserved in case of a constant source vector.

If $w^T f(y) \leq 0$, i.e. $w^T y(t)$ decreases with time, it follows from (D.2) that $w^T y^{n+1} \leq w^T y^n$, provided that $w^T Y^n \leq w^T y^n$. In that case, for the modified Newton iterates the relation

$$w^T y^{(l)} \leq w^T y^n, \quad l \geq 1, \tag{D.5}$$

holds, even if (D.5) does not hold for the start vector $y^{(0)}$. This can be seen from (D.4) using the assumption $w^T Y^n \leq w^T y^n$. This assumption need not be valid for general BDF methods. For BDF1 it is valid because then $Y^n = y^n$. For BDF2 it is also valid. The history vector for variable step sizes can be written as $Y^n = y^n + \alpha(y^n - y^{n-1})$, with $\alpha > 0$. If we assume that the solution y^n of the last time step satisfies the condition $w^T(y^n - y^{n-1}) \leq 0$, it follows that the assumption is valid. For BDF3 the assumption is not valid in general.

In the above proof, the Jacobian matrix J plays a crucial role. The only condition that needs to be satisfied is $w^T J(y) = 0$ for all y. If the Jacobian is obtained by numerical differencing, this relation may not hold. Of course, also the linear system (D.4) needs to be solved exactly.

Appendix E

Coefficients for 3STEP

We write the third order BDF formula in the form

$$y^{n+1} = \beta_n y^n + \beta_{n-1} y^{n-1} + \beta_{n-2} y^{n-2} + \gamma\tau f(y^{n+1}).$$

The parameters β_i and γ depend on two step size ratios

$$c_0 = \frac{t^n - t^{n-1}}{t^{n+1} - t^n}, \quad c_1 = \frac{t^{n-1} - t^{n-2}}{t^{n+1} - t^n}.$$

Solving the order conditions gives

$$\begin{aligned}
\beta_{n-2} &= \frac{(1+c_0)^2}{\alpha c_1 (c_0 + c_1)}, \\
\beta_{n-1} &= \frac{-(1+c_0+c_1)^2}{(\alpha+1)c_0 c_1}, \\
\beta_n &= 1 - \beta_{n-1} - \beta_{n-2}, \\
\alpha &= 2c_1 + c_0 c_1 + 4c_0 + c_0^2 + 3,
\end{aligned}$$

and the parameter γ is given by

$$\gamma = \beta_n + (1+c_0)\beta_{n-1} + (1+c_0+c_1)\beta_{n-2}.$$