Centrum voor Wiskunde en Informatica
**REPORT***RAPPORT*

# An implicit-explicit approach for atmospheric transport-chemistry problems

J.G. Verwer, J.G. Blom and W. Hundsdorfer

Department of Numerical Mathematics

**NM-R9501 1995**

Report NM-R9501
ISSN 0169-0388

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

# An Implicit-Explicit Approach for Atmospheric Transport-Chemistry Problems

J.G. Verwer, J.G. Blom and W. Hundsdorfer

*CWI*

*P.O. Box 94079, 1090 GB Amsterdam, The Netherlands*

**Abstract**

We investigate numerical algorithms for use in air pollution models. The emphasis lies on time integration aspects in connection with advection, vertical turbulent diffusion and stiff chemical transformations. The time integration scheme considered is a 2nd-order implicit-explicit BDF scheme which handles advection explicitly and vertical turbulent diffusion and chemistry implicitly and coupled. The investigation is divided into three parts. In the first part we propose a Gauss-Seidel technique for the implicit solution of the chemistry and vertical turbulent diffusion. For a hypothetical 1D model, based on a 66-species EMEP photochemical ozone chemistry scheme, this technique is shown to be significantly more efficient than the usual approach of using modified Newton with a linear band solver. For the stiff chemistry the Gauss-Seidel iteration is effectively explicit. For the diffusion the implicitness is retained, which gives rise to tridiagonal linear systems. In the second part we discuss stability and consistency properties of the implicit-explicit BDF scheme, assuming the 3rd-order upwind biased finite-difference discretization of the advection operator. In the third part we apply the implicit-explicit scheme to a hypothetical 3D model based on the same photochemical ozone chemistry. Here we employ vectorization over the horizontal grid. Grid-vectorization is of large practical interest as it leads to significantly higher efficiency on a vector computer. Dependent of the process at hand, parallelization is obtained either over the horizontal or over the vertical grid.

*AMS Subject Classification (1991):* Primary: 65M06, 65M20. Secondary: 65Y05, 65Y20.
*CR Subject Classification (1991):* G.1.8, G.1.1, J.2
*Keywords & Phrases:* long range transport air pollution models, time-dependent advection-diffusion-reaction, numerical methods, vectorization, parallelization.
*Note:* This report is one of a series on the development of algorithms for long range transport air pollution models. We gratefully acknowledge support from the RIVM (projects EUSMOG and CIRK) and from Cray Research Inc. under grant CRG 94.01 via the Stichting Nationale Computer Faciliteiten (National Computing Facilities Foundation, NCF).

## 1. Introduction

Atmospheric transport-chemistry problems used in air pollution modeling take into account many physical processes. From the numerical point of view important processes are the stiff chemical transformations, advective transport, caused by horizontal wind mainly, and vertical transport caused by turbulent diffusion. In air pollution models the combination of transport and chemistry is important since many polluting species have reaction times large enough to be transported over long distances before they have disappeared. This paper is devoted to a numerical study of a 3D transport-chemistry model of which the governing partial differential equation is given by

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\underline{u}\rho) = \frac{\partial}{\partial \sigma}(K\frac{\partial \rho}{\partial \sigma}) + r(t,\rho), \tag{1.1}$$

in spherical coordinates. Here, $\rho = \rho(\lambda, \phi, \sigma, t)$ is a vector in $\Re^m$ of $m$ concentration values, $\underline{u} = ue_\lambda + ve_\phi$ is the horizontal velocity vector with $(e_\lambda, e_\phi)$ the unit vectors on the sphere in the longitude

($\lambda$) and latitude ($\phi$) direction, $K$ is a scalar diffusion coefficient in the vertical direction ($\sigma$), and the horizontal divergence operator is given by [20]

$$\mathrm{div}(\underline{u}\rho) = \frac{1}{\alpha \cos\phi}\left(\frac{\partial}{\partial\lambda}(u\rho) + \frac{\partial}{\partial\phi}(v\rho\cos\phi)\right), \tag{1.2}$$

where $\alpha$ is the radius of the earth. Vertical advection and horizontal diffusion can be added without essential numerical difficulties. Note that in reality these two transport processes are significantly less important than horizontal advection and vertical turbulent diffusion. The vector function $r(t,\rho)$ defining the chemical transformations, emission and dry deposition, has the special form

$$r(t,\rho) = P(t,\rho) - L(t,\rho)\rho, \tag{1.3}$$

where $P(t,\rho)$ is the vector of production terms and $L(t,\rho)\rho$ the vector of loss terms with $L(t,\rho)$ a diagonal matrix. For many species the reciprocal of their entry in $L$ is a good approximation of the physical time constant or characteristic reaction time. In virtually all applications the range of reaction times is so large that we have to face the difficulty of stiffness [12].

Our numerical study concerns a direct scheme which is derived along the method of lines (MOL). The integration method used is based on a 2nd-order implicit-explicit backward differentiation formula (BDF) which handles advection explicitly and chemistry and vertical diffusion implicitly and coupled. The numerical study has three focus points. The first is taken up in Section 2 and concerns the coupled solution of chemical transformations and vertical turbulent diffusion. The chemistry computation is of major importance since in practice this computation often turns out to dominate CPU costs. CPU costs can be extremely high for advanced 3D air pollution models containing a large number of species, which necessitates the use of very fast, special purpose solvers. Normally, for stability reasons stiffness impedes the use of some form of Newton iteration for computing the implicitly defined solutions. In [18, 19] we have shown, for different box models and using the implicit 2nd-order BDF formula, that for atmospheric chemistry problems the more simple Gauss-Seidel iteration can be used with greater efficiency. In fact, the Gauss-Seidel method used is truly explicit and is related to simple, explicit quasi-steady-state-approximation (QSSA) schemes to which it compares very favorably as well [19]. An additional advantage is the much lower memory demand compared to a Newton method. In this paper we show that the Gauss-Seidel iteration from [18, 19] can be effectively extended to the coupled chemistry-diffusion case.

In Section 3 the second point is addressed, viz. the consistency and stability in time of the implicit-explicit MOL scheme, given that the advection operator is discretized by the mass-conservative, flux-limited finite-difference scheme proposed in [8]. This finite-difference scheme is based on the 3rd-order upwind biased discretization (the $\kappa = 1/3$-scheme in the terminology of van Leer) and has been found to be very suitable for our application (see also [1, 3]). We give results of a Fourier von Neumann analysis which show that the explicit, two-step advection scheme yields time step stability limits sufficiently large for the application of the complete implicit-explicit scheme. Because the use of the Gauss-Seidel method renders the diffusion-chemistry computation also explicit, except for tridiagonal matrix inversions, the combined implicit-explicit approach thus results in an efficient, almost explicit process for stiff models of type (1.1).

Section 4 deals with the third point, which concerns the accuracy and efficiency performance of the implicit-explicit MOL scheme when applied to actual 3D problems. Efficiency is of utmost importance and since we currently use a Cray C98/4256 (4 vector processors, shared memory), special attention will be paid to vectorization and parallelization of the advection and diffusion-chemistry computation in 3D applications. For the latter, vectorization over the horizontal grid, as advocated in [9] for atmospheric problems and in [16] for shallow-water transport problems, leads to significant increase

in computational efficiency. Parallelization is obtained mainly over the vertical grid. In the tridiagonal matrix inversion both parallelization and vectorization are done over the horizontal grid points.

## 2. The Gauss-Seidel process

For examining the Gauss-Seidel process it is convenient to omit the advection part. For the time being we thus consider the general 1D diffusion-chemistry problem

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial \sigma}(K\frac{\partial \rho}{\partial \sigma}) + r(t,\rho), \quad t > t_0, \quad 0 < \sigma < \sigma_H, \tag{2.1}$$

supplemented with the initial condition $\rho(\sigma, t_0) = \rho^0(\sigma)$ and the boundary conditions

$$(K\frac{\partial \rho}{\partial \sigma})(0,t) = 0, \quad (K\frac{\partial \rho}{\partial \sigma})(\sigma_H, t) = 0. \tag{2.2}$$

### *2.1 The discretization schemes*

The vertical turbulent diffusion term is discretized on the non-equidistant, cell-centered grid

$$\Omega_V = \{\sigma_k : \sigma_1 = \tfrac{1}{2}\Delta\sigma_1,\ \sigma_k = \sigma_{k-1} + \tfrac{1}{2}(\Delta\sigma_{k-1} + \Delta\sigma_k),\ 2 \le k \le N_\sigma\}, \tag{2.3}$$

such that the following ODE system is obtained,

$$\frac{d}{dt}c_k = f_k(t,c) \doteq d_k(t,c) + r(t,c_k), \quad t > t_0, \quad c_k(t_0) = \rho^0(\sigma_k), \quad 1 \le k \le N_\sigma. \tag{2.4}$$

Here $c(t)$ is the complete grid function on $\Omega_V$, $c_k(t) \approx \rho(\sigma_k, t)$ and

$$d_k(t,c) = \frac{8K_k^+ \frac{c_{k+1}-c_k}{\Delta\sigma_{k+1}+\Delta\sigma_k} - 8K_k^- \frac{c_k - c_{k-1}}{\Delta\sigma_k + \Delta\sigma_{k-1}}}{\Delta\sigma_{k+1} + 2\Delta\sigma_k + \Delta\sigma_{k-1}}, \quad 1 \le k \le N_\sigma, \tag{2.5}$$

with $K_k^{\pm} = K(t, (\sigma_k + \sigma_{k\pm1})/2)$ and $\Delta\sigma_0 = \Delta\sigma_1, \Delta\sigma_{N_\sigma+1} = \Delta\sigma_{N_\sigma}$. Note that $K$ is evaluated half way between the cell centers, rather than at the cell boundaries, to obtain a consistent discretization of (at least) order one. The boundary conditions are incorporated by putting $K_k^- = 0$ for $k = 1$ and $K_k^+ = 0$ for $k = N_\sigma$. Also note that $c_k(t)$ is a vector in $\Re^m$ and that the diffusion operator introduces no coupling between different species. For each of the species the semi-discrete diffusion operator is equal and constituted by the same tridiagonal matrix. The species are coupled through the chemistry term $r(t, c_k)$.

For time integration we use the familiar 2nd-order, two-step implicit BDF formula in variable step form,

$$c_k^{n+1} = C_k^n + \gamma\tau f_k(t_{n+1}, c^{n+1}), \quad 1 \le k \le N_\sigma, \quad n \ge 1, \tag{2.6}$$

where $c_k^n \approx c_k(t_n)$ and

$$C_k^n = \left((1+q)^2 c_k^n - q^2 c_k^{n-1}\right)/(1+2q), \tag{2.7}$$

$\tau = t_{n+1} - t_n, \gamma = (1+q)/(1+2q)$ and $q = (t_{n+1} - t_n)/(t_n - t_{n-1})$. The initial vector $c_k^0 \doteq c_k(t_0)$ and $c_k^1$ is supposed to be defined by the 1st-order implicit Euler rule. This combination yields 2nd-order accurate time stepping which for atmospheric transport applications is sufficient in view of the modest accuracy requirement. Generally, relative accuracy larger than 1% is superfluous.

*2.2 Gauss-Seidel iteration*

Suppressing the temporal index $n+1$ and $t_{n+1}$ for notational convenience, we write (2.6) as

$$c_k = C_k^n + \gamma\tau d_k(c) + \gamma\tau P(c_k) - \gamma\tau L(c_k)c_k, \quad 1 \le k \le N_\sigma. \tag{2.8}$$

Let $c_k^{(j)}$ be the $j$-th component of $c_k$ and introduce, for each of the $m$ species, the following vectors on $\Omega_V$,

$$\mathbf{c}^{(j)} = [c_1^{(j)}, \ldots, c_{N_\sigma}^{(j)}]^T, \quad \mathbf{P}^{(j)}(\mathbf{c}) = [P^{(j)}(c_1), \ldots, P^{(j)}(c_{N_\sigma})]^T, \quad j = 1, \ldots, m. \tag{2.9}$$

The vector $\mathbf{c}$ is supposed to contain all vectors $\mathbf{c}^{(j)}$. Assume a similar definition for the vector $\mathbf{C}^{(j)}$ of known values and introduce the diagonal matrices

$$\mathbf{L}^{(j)}(\mathbf{c}) = \mathrm{diag}(L^{(j)}(c_1), \ldots, L^{(j)}(c_{N_\sigma})), \quad j = 1, \ldots, m. \tag{2.10}$$

Then we may write

$$\mathbf{c}^{(j)} = \mathbf{C}^{(j)} + \gamma\tau\mathbf{A}\mathbf{c}^{(j)} + \gamma\tau\mathbf{P}^{(j)}(\mathbf{c}) - \gamma\tau\mathbf{L}^{(j)}(\mathbf{c})\mathbf{c}^{(j)}, \quad j = 1, \ldots, m, \tag{2.11}$$

where $\mathbf{A}$ is the tridiagonal (diffusion) matrix of order $N_\sigma$ (cf. (2.5)). Equivalently we have

$$\mathbf{c}^{(j)} = \left(\mathbf{I} - \gamma\tau\mathbf{A} + \gamma\tau\mathbf{L}^{(j)}(\mathbf{c})\right)^{-1} \left(\mathbf{C}^{(j)} + \gamma\tau\mathbf{P}^{(j)}(\mathbf{c})\right), \quad j = 1, \ldots, m, \tag{2.12}$$

since the inverse of the tridiagonal matrix $\mathbf{I} - \gamma\tau\mathbf{A} + \gamma\tau\mathbf{L}^{(j)}(\mathbf{c})$ always exists. Note that the matrix would be symmetric on a uniform grid. In general it is non-symmetric though, since we will use a non-uniform grid $\Omega_V$.

The Gauss-Seidel iteration process for solving $\mathbf{c}^{(j)}, 1 \le j \le m$, is carried out on equation (2.12) and consists of the following calculations. Let $\mathbf{c}_{[i]}$ denote the $i$-th iterate for $\mathbf{c}$. Then,

1. Initial estimation: $i = 0$, $\mathbf{c}_{[i]} = \max(0, \mathbf{c}^n + q(\mathbf{c}^n - \mathbf{c}^{n-1}))$.
   2. Compute, in the order $j = 1, \ldots, m$ :
      $2a$. $\mathbf{L}^{(j)}(\mathbf{c}_{[i]}), \mathbf{P}^{(j)}(\mathbf{c}_{[i]})$.
      $2b$. LU-decompose $\mathbf{I} - \gamma\tau\mathbf{A} + \gamma\tau\mathbf{L}^{(j)}(\mathbf{c}_{[i]}))$. (2.13)
      $2c$. Backsolve $\mathbf{c}_{[i+1]}^{(j)}$.
      $2d$. Update $\mathbf{c}_{[i]} = (\mathbf{c}_{[i+1]}^{(1)}, \ldots, \mathbf{c}_{[i+1]}^{(j)}, \mathbf{c}_{[i]}^{(j+1)}, \ldots, \mathbf{c}_{[i]}^{(m)})$.
3. Put $i := i + 1$. If more iterations are required, then go to 2.

Note that the approximations are corrected specieswise and simultaneously over the grid. Hence, the diffusion term is treated implicitly. This requires the tridiagonal matrix calculations $2b, 2c$ any time a species is corrected. Thus, except for the tridiagonal matrix calculations, the Gauss-Seidel process is truly explicit. No Jacobian matrices for the chemistry system are computed and no additional storage is required. Would there be no diffusion, then this process is completely identical to that used for the box models in [18, 19]. On the other hand, without chemistry the diffusion is treated implicitly in the usual way. This means that method (2.13) differs from the well-known classical nonlinear Gauss-Seidel method, since this classical method does not distinguish between the diffusion and reaction terms and would be applied directly to (2.8). In fact, also without diffusion (2.13) slightly differs from the classical method when applied to (2.8). For box models the two are equivalent if the $j$-th entry of the diagonal matrix $L(t, \rho)$ does not depend on $\rho^{(j)}$. Note that for the production vector $P(t, \rho)$ this always holds. In [11] the classical Gauss-Seidel technique for solving the linear system of equations arising in the modified Newton iteration is discussed for chemical kinetics problems.

*2.3 A variable stepsize GS and MN implementation*

In the experiments in Section 2.6 variable stepsizes are used. The stepsize strategy is similar as in [18, 19] and many stiff ODE codes and identical for Gauss-Seidel (GS) and modified Newton (MN) iteration. To save space we omit details here. We have not implemented an iteration strategy for GS-iteration and hence prescribe the number of GS-iterations beforehand. To our experience this works well using only a few iterations. The MN iteration strategy slightly differs from the standard approach. New Jacobians are only computed if the convergence of the Newton process is not fast enough. This is different from strategies used in stiff ODE solvers where Jacobians and/or decompositions are also computed if the time step size changes too much. So, as a rule, we do not compute a new decomposition if the stepsize changes, unless this coincides with slow convergence. This strategy appeared to be the most efficient in our experiments. We use an analytical Jacobian matrix for the chemistry part (evaluated with MAPLE). This matrix is sparse which means that Jacobian evaluations are cheap. A standard LINPACK [5] linear band solver is used for the block-tridiagonal linear system. We note that for this system the fill-in of the matrix factorization is almost complete, ruling out a sparse system solver.

*2.4 The 1D test problem*

Our test problem (2.1)-(2.2) is based on the state-of-the-art EMEP MSC-W ozone chemistry (140 reactions between 66 species [14, 15]). In [19] we have used the same chemistry in box model tests. The experiments reported here extend these to column model tests in a straightforward way. This means that with the exception of a variable mixing height, we use at any of the vertical grid points the same chemical reactions, reaction constants, photolysis rates, temperature and other relevant meteorological parameters. Note that photolysis rates undergo a discontinuity at sunset and sunrise. We use the 'rural case' emission input from [19] at a fixed part of the air column near the ground (ca. $40m$.). Above this height emissions are switched off. We emphasize that the actual emission input used near the ground are a factor 30 larger than in [19], since there the emission was uniformly distributed over the total air column of length 1200 $m$. Consequently, near the ground the chemistry is more difficult to solve since larger concentration values and concentration gradients arise, in spite of the vertical distribution caused by the turbulent diffusion term.

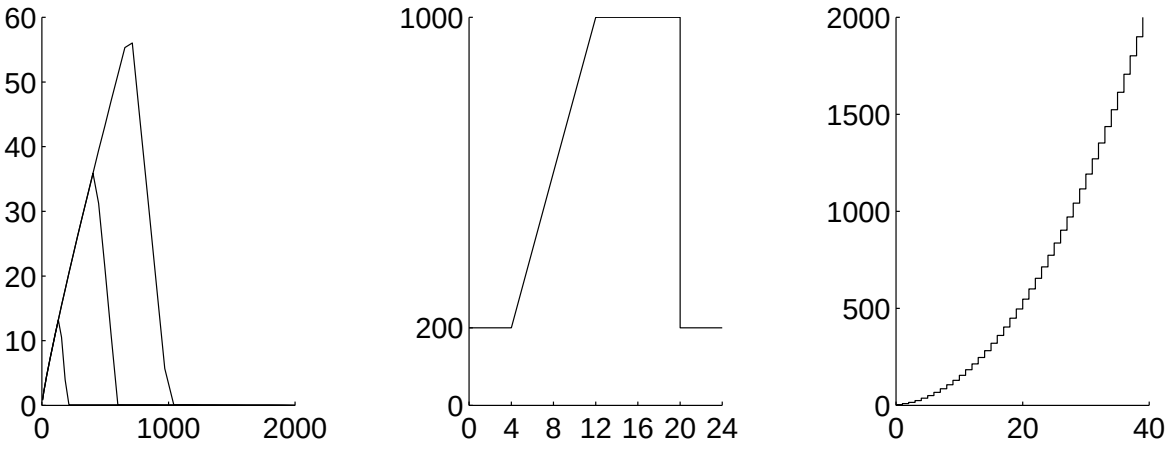


Figure 1: The diffusion coefficient $K(t,\sigma)$ in the unit $m^2/sec.$ for $t = 04.00h, 08.00h, 12.00h$ (left plot); the mixing height (middle plot) and the non-equidistant space grid (right plot).

The turbulent diffusion coefficient $K(t,\sigma)$ depends on the mixing height which depends on the time of day. In Figure 1 we have plotted $K(t,\sigma)$ for some values of time upon to the height of $\sigma_H = 2000\,m$.

Figure 1 also shows the mixing height (sunrise at 04.00 *hours*, sunset at 20.00 *hours*) and the non-equidistant space grid which contains 40 points. In applications non-equidistant grids are used since above the mixing height the turbulent diffusion is negligibly small, whereas near the ground small grid sizes are required to more accurately resolve the ground emission (see also [7, 10, 13]). Because we focus on time integration efficiency, we here use one fixed space grid (see Figure 1) and use the ODE error on this grid in the comparison in Section 2.6. Hence we pay no separate attention to the spatial accuracy, which is approximately 1% in the error norm used in Section 2.6. Note that for our grid the dimension of the banded linear system arising in MN iteration is $mN_\sigma = 2640$ with a bandwidth equal to $2m + 1 = 133$.

A total time interval of 112 hours is used which starts at sunrise at day one ($t = 14400\,sec.$) and ends at sunset at day five ($t = 417600\,sec.$). Note, however, that in all integrations the 112 hour interval is divided into 56 two hour intervals, at each of which we restart the integration with the one-step backward Euler formula using a tenfold smaller stepsize. This division into 56 subintegrations was also used in [19] and is in accordance with regular changes in model coefficients and input. Such changes can introduce a discontinuity (like at sunset and sunrise), motivating the many restarts. Observe that when the current procedures would be used in an operator splitting scheme, that then also frequent restarts are to be made.

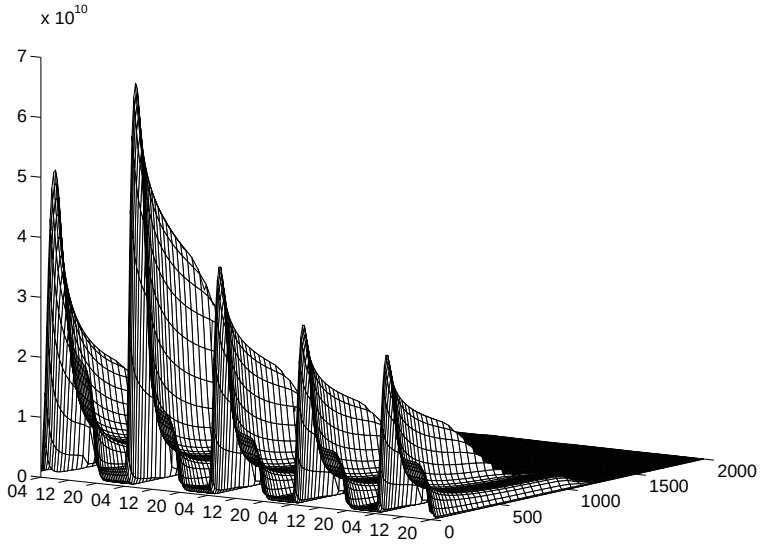


Figure 2: The 1D reference solution representing the diurnal variation of the $NO$ concentration along the column. Unit is the number of molecules per $cm^3$.

As initial function $\rho^0(\sigma)$ an arbitrary initial guess is used (cf. [19]). This initial guess introduces strong initial transients which have disappeared completely after one day. Figure 2 shows the time evolution of the component representing the $NO$ column over the complete 112 hour time interval (this figure is the counterpart of Figure 2 in [19]). The solution plotted is a very accurately computed reference solution. Although we have not attempted to simulate a 'real atmosphere' in all respects, the figure nicely shows the day-night transitions and the space dependency due to the vertical diffusion. Notice that in reality the boundary condition (2.2) at the bottom often contains a term connected with dry deposition. Similar as in [19], we have incorporated dry deposition in the rate equations, viz. in the same part of the air column near the ground as used for the ground level emissions. Other

types of boundary conditions are numerically easy to incorporate.

*2.5 The stiffness*

The stiffness in the chemistry can be estimated from the characteristic reaction times $L^{(j)^{-1}}$ [12]. For a certain time interval during the day a rough guess is $10^7 sec.$ as the maximum and $10^{-9} sec.$ as the minimum time constant. This obviously reveals excessive stiffness. The stiffness induced by the turbulent diffusion term is difficult to estimate due to the nonuniformity of the grid and the variability of $K$. However, it is quite modest in comparison with the chemistry stiffness. Near the ground the smallest grid cells used vary in length between $1\,m.$ and $10\,m.$, approximately. In view of the values $K$ may take, we thus estimate that the minimal values for the characteristic diffusion times $(\Delta\sigma_k)^2/K$ vary between 1 and 10 $sec.$ This, however, is still small in comparison with the $7200\,sec.$ time interval length in view of the fact that stepsizes in the order of a few minutes and larger should be achievable for most of the time.

*2.6 The experiments*

Two different experiments have been carried out. The first experiment serves to provide insight in the accuracy-efficiency performance of GS iteration when varying $RTol$ (the relative tolerance parameter for the variable stepsize selection) and the number of GS iterations. The second experiment serves to compare GS iteration with MN iteration. Efficiency is measured by $CPU$ time and accuracy by the number of correct digits

$$SDA = -log_{10}\left(\frac{1}{66}\sum_{j=1}^{66}\left[\sum_{n=8}^{56}\sum_{k=1}^{40}(sol_{kj}^n - app_{kj}^n)^2\right]^{\frac{1}{2}} \Big/ \left[\sum_{n=8}^{56}\sum_{k=1}^{40}(sol_{kj}^n)^2\right]^{\frac{1}{2}}\right), \tag{2.14}$$

where $sol_{kj}^n$ denotes a highly accurate approximation to the ODE solution on the grid and $app_{kj}^n$ the numerical solution. The times $t_n$ are restricted to $t_n = 14400 + 7200n$ with $8 \leq n \leq 56$, $j$ runs over all species, and $k$ runs over the grid. Hence we sample at the end of each 2 hour interval, but for the first time at sunset at the first day because we start with an arbitrary initial condition. We thus assume to have fully get rid of the initial transients at the first sunset. Note that we use an $l_2$ error in time and over the grid for each species and that we then average over all species.

Figure 3a gives results of the first experiment which comprises 16 integrations. Each of the 4 lines corresponds with a prescribed numbers of GS iterations ($\circ$-2, $\times$-4, +-6, $*$-8) and connects results for four values of $RTol$ ($10^{-1}$, $10^{-2}$, $10^{-3}$, $10^{-4}$). This enables us to compare the use of different values of $RTol$ and a fixed number of GS iterations with the use of a different number of GS iterations and a fixed value of $RTol$. First we notice that decreasing $RTol$ by the chosen factor of 10, also reduces the error with this same factor, approximately. This indicates that the variable stepsize strategy works well in this respect. We also see that increasing the number of GS iterations from 2 to 8 also results in an approximately 10 times smaller error which explains why the 4 lines almost coincide. These gain factors resulting from increasing the number of GS iterations seem more or less characteristic for our application, although as yet we cannot support this with more evidence. However, to our experience a good strategy is to keep the number of GS iterations low and to take $RTol$ small, rather than using a large number of iterations and a crude tolerance. Then we do not need a GS-iteration strategy and we do not risk a number of GS iterations larger than the amount needed to reach the accuracy of the implicit 2-nd order BDF method.

Figure 3b gives results of the second experiment. Accuracy is plotted against efficiency for 4 integrations using $RTol = 10^{-1}, 10^{-2}, 10^{-3}$ and $10^{-4}$, both for GS-iteration and MN-iteration. Here we have fixed the number of GS-iterations to 4. Notice that the number of Jacobian updates, backsolves,

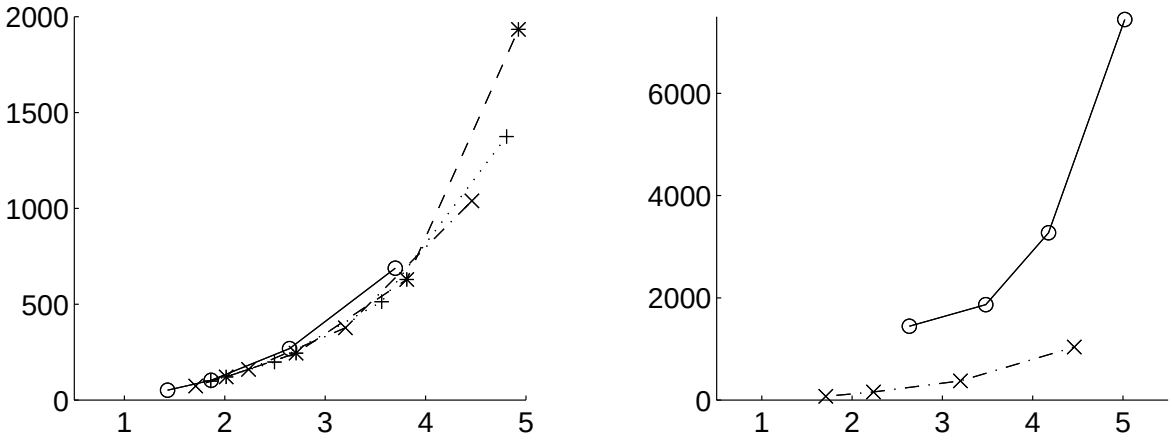


Figure 3: 1D experiment, CPU*sec* vs. SDA: (a) GS-iteration, (b) GS-iteration(×) versus MN-iteration(∘).

etc. in the MN-iteration is determined automatically by the implemented strategy. In line with the results of the first experiment, we see that MN-iteration always results in smaller errors (close to the error of the BDF formula), but clearly at the expense of much higher costs. GS iteration appears to be 4 to 5 times more efficient. This strongly favors GS iteration, certainly so for 3D problems where 1D calculations of the type considered here need to be carried out at thousands of points in an horizontal grid. An example will be given in Section 4.

## 3. Consistency and stability of the implicit-explicit scheme

In this section we address our second subject, viz. the consistency and stability in time of the implicit-explicit two-step BDF scheme for (1.1). We hereby assume that the spherical advection operator (1.2) is discretized on some cell-centered uniform grid

$$\Omega_H = \{(\lambda_i, \phi_j) : \lambda_i = (i - \tfrac{1}{2})\Delta\lambda, \phi_j = (j - \tfrac{1}{2})\Delta\phi\},$$

by means of the mass-conservative, flux-limited, finite-difference scheme proposed in [8], Section 5.3. This scheme is based on the 3rd-order upwind biased method which is equivalent to the $\kappa = 1/3$-scheme of van Leer and derived with the aim to produce positive and monotone solutions and little artificial diffusion. In [8] it has been found to be very suitable for our application (see also [1, 3]). To save space we refer to [8] for the actual formulas and their derivation.

### *3.1 The 2nd-order implicit-explicit BDF scheme*

Let $c(t)$ denote the semi-discrete grid function on the cell-centered 3D grid $\Omega_H$ x $\Omega_V$ with components $c_{ijk}(t)$, now approximating $\rho$ at the grid point $(\lambda_i, \phi_j, \sigma_k)$. Let

$$\frac{d}{dt}c = g(t, c) + f(t, c) \tag{3.1}$$

denote the associated semi-discrete 3D problem. Components $f_{ijk}$ are defined by (2.4) and components $g_{ijk}$ represent the numerical advection scheme (boundary conditions for the advection operator are omitted, for the time being). Recall that each component $c_{ijk}$ itself is a vector in $\Re^m$. Also recall that $f(t, c)$ is only coupled in $\Omega_V$ and $g(t, c)$ only in $\Omega_H$. Further, since in the advection operator species

are not coupled to one another, we might consider the semi-discrete advection system for each species separately.

Assuming constant stepsizes, for simplicity of presentation, the two-step implicit BDF formula applied to (3.1) reads

$$c^{n+1} = \frac{4}{3}c^n - \frac{1}{3}c^{n-1} + \frac{2}{3}\tau g(t_{n+1}, c^{n+1}) + \frac{2}{3}\tau f(t_{n+1}, c^{n+1}), \quad n \geq 1. \tag{3.2}$$

However, we do not wish to integrate (3.1) implicitly and therefore replace (3.2) by the implicit-explicit scheme

$$c^{n+1} = \frac{4}{3}c^n - \frac{1}{3}c^{n-1} + \frac{2}{3}\tau g(t_{n+1}, 2c^n - c^{n-1}) + \frac{2}{3}\tau f(t_{n+1}, c^{n+1}), \quad n \geq 1. \tag{3.3}$$

Likewise, for $n = 0$ the implicit Euler rule is replaced by the 1st-order implicit-explicit Euler rule

$$c^1 = c^0 + \tau g(t_0, c^0) + \tau f(t_1, c^1). \tag{3.4}$$

Integration schemes of this type are well-known [2, 17]. For our application (3.3) is particularly attractive as the method is only 1D implicit. One step with (3.3) amounts to computing advection explicitly at all horizontal grids $\Omega_H$ and vertical diffusion and chemistry implicitly and coupled along all vertical grids $\Omega_V$ perpendicular to $\Omega_H$, as discussed in Section 2.

The implicit-explicit approach may be viewed as splitting within a method. The additional error introduced by 'internally splitting' advection from diffusion and chemistry preferably should be of the same size as the error of the original BDF formula. Substitution of a sufficiently differentiable solution $c(t)$ into (3.3) yields the local expansion

$$\begin{aligned} c(t_{n+1}) &= \frac{4}{3}c(t_n) - \frac{1}{3}c(t_{n-1}) + \frac{2}{3}\tau g(t_{n+1}, 2c(t_n) - c(t_{n-1})) + \frac{2}{3}\tau f(t_{n+1}, c(t_{n+1})) \\ &+ \frac{2}{9}\tau^3 \frac{d^3}{dt^3}c(t_n) + \frac{2}{3}\tau^3 g'(t_n, c(t_n))\frac{d^2}{dt^2}c(t_n) + O(\tau^4). \end{aligned} \tag{3.5}$$

Hence the extrapolation in (3.3) adds the product of the Jacobian $g'$ with the 2nd derivative to the original local truncation error, but the local error remains $O(\tau^3)$. To retain the level of local accuracy, the size of this product thus should compare with the size of the 3rd solution derivative. Alternatively, we can examine the local temporal accuracy of the implicit-explicit formula for the PDE itself, by directly applying the integration formula to (1.1). The local truncation error expression then becomes

$$\begin{aligned} &\frac{2}{9}\tau^3\rho_{ttt}(t_n) + \frac{2}{3}\tau^3 \mathrm{div}(\underline{u}[\rho(t_{n+1}) - 2\rho(t_n) + \rho(t_{n-1})]) + O(\tau^4) \\ &= \frac{2}{9}\tau^3\rho_{ttt}(t_n) + \frac{2}{3}\tau^3 \mathrm{div}(\underline{u}\rho_{tt}(t_n)) + O(\tau^4). \end{aligned} \tag{3.6}$$

We see that if the 3rd temporal derivative is of the same size as the divergence of the velocity times the 2nd temporal derivative, then no reduction in local accuracy will result. This observation is of relevance only when the spatial error is negligibly small. Of course, if this error dominates, then 'internal splitting' never will harm accuracy.

*3.2 Linear stability*

Treating advection explicitly obviously has a large impact on stability. To examine this we consider the linear, constant coefficient system

$$\rho_t + u\rho_\lambda = K\rho_{\sigma\sigma} + M\rho, \tag{3.7}$$

where $M$ is a matrix representing the stiff chemistry. Note that for the purpose of linear stability analysis it suffices to consider only 1D-advection. Conclusions for 2D (and 3D) advection can immediately be drawn from the 1D analysis. By assuming that $M$ is similar to its diagonal eigenvalue matrix, we may as well study the componentwise equations arising in the eigensystem expansion. Using the same notation, we thus proceed with the scalar equation

$$\rho_t + u\rho_\lambda = K\rho_{\sigma\sigma} + \mu\rho. \tag{3.8}$$

We will apply the Fourier method of von Neumann and thus tacitly assume that the $\kappa$-scheme is applied without limiting to maintain the linearity and that $\Omega_V$ is uniform. For $u > 0$, e.g., this results in the semi-discrete scheme

$$\frac{d}{dt}c_{ik} + u\frac{c_{i-2k} - 6c_{i-1k} + 3c_{ik} + 2c_{i+1k}}{6\Delta\lambda} = K\frac{c_{ik-1} - 2c_{ik} + c_{ik+1}}{(\Delta\sigma)^2} + \mu c_{ik}. \tag{3.9}$$

Carrying out the Fourier analysis leads to the familiar characteristic equation

$$(1 - \frac{2}{3}\tau(A_\sigma + \mu))\alpha^2 - (\frac{4}{3} + \frac{4}{3}\tau A_\lambda)\alpha + (\frac{1}{3} + \frac{2}{3}\tau A_\lambda) = 0, \tag{3.10}$$

where $A_\lambda$ and $A_\sigma$ are the advection and diffusion eigenvalues,

$$A_\lambda = -\frac{1}{3}\frac{u}{\Delta\lambda}\left(\sqrt{-1}\,\sin\theta_\lambda(4 - \cos\theta_\lambda) + \mathrm{sign}(u)(1 - \cos\theta_\lambda)^2\right), \quad \theta_\lambda = \omega_\lambda\Delta\lambda, \tag{3.11}$$

$$A_\sigma = \frac{2K}{\Delta\sigma^2}(\cos\theta_\sigma - 1), \quad \theta_\sigma = \omega_\sigma\Delta\sigma. \tag{3.12}$$

for all $|\theta_\lambda|, |\theta_\sigma| \leq \pi$. We note in passing that when the term $\mathrm{sign}(u)(1 - \cos\theta_\lambda)^2$ is removed, the advection eigenvalue $A_\lambda$ for the classical 4-th order central difference scheme will result [8]. Then $A_\lambda$ is purely imaginary. In the upwind case $A_\lambda$ is complex with a negative real part.

For investigating the root condition we consider first the stability region $S$ of the explicit two-step scheme contained in (3.3),

$$c^{n+1} = \frac{4}{3}c^n - \frac{1}{3}c^{n-1} + \frac{2}{3}\tau g(t_{n+1}, 2c^n - c^{n-1}). \tag{3.13}$$

The region $S$ is easily determined with the root locus curve computation [6]. If $\tau A_\lambda \in S$ for all $\theta_\lambda$, then we have von Neumann stability for the explicit advection scheme. In Figure 4 we have plotted the boundary of $S$ and the curve $\tau A_\lambda$, as a function of $\theta_\lambda$, for three trial-and-error values of the Courant number

$$\nu_\lambda = \frac{\tau|u|}{\Delta\lambda}, \tag{3.14}$$

viz., for $\nu_\lambda = 0.3, 0.45, 0.6$. The figure shows that the explicit two-step scheme is stable and that the maximal CFL number is 0.45, approximately. In higher space dimension the analysis goes entirely similar. For example, for

$$c_t + uc_\lambda + vc_\phi = 0, \tag{3.15}$$

we must have $\tau(A_\lambda + A_\phi) \in S$ for all frequencies. Here $A_\phi$ is the 'advection eigenvalue' defined similarly as $A_\lambda$. A safe upperbound for the stepsize $\tau$ of the explicit advection scheme then is obtained from

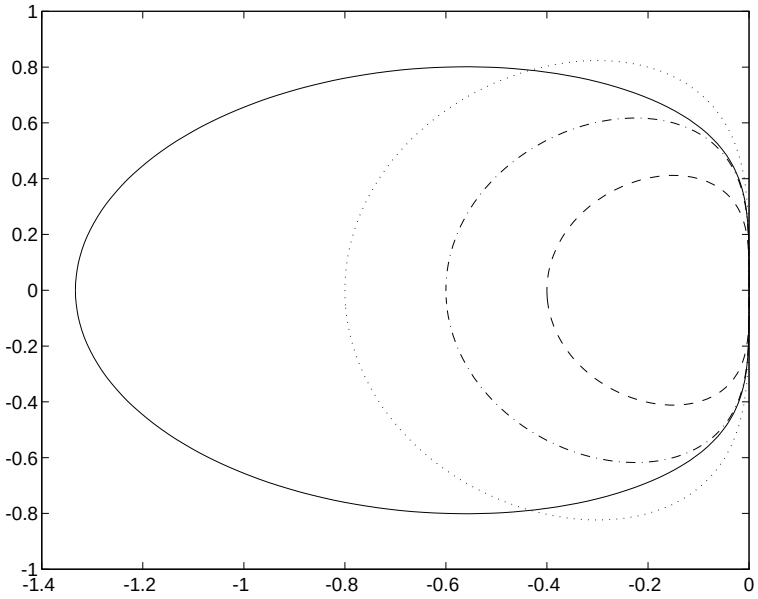


Figure 4: The stability region $S$ of the explicit two-step scheme (3.13) (solid line). The dashed, dashdotted and dotted line are the curves for $\tau A_\lambda$ for the CFL numbers $\nu = 0.3, 0.45, 0.6$, respectively.

$$\tau \left( \frac{|u|}{\Delta\lambda} + \frac{|v|}{\Delta\phi} \right) \leq 0.45. \tag{3.16}$$

Note that for central differences the explicit advection scheme will be unstable for all $\tau > 0$, because the eigenvalues then lie on the imaginary axes which is not intersected by $S$. In the case of a central scheme one should start from a different implicit method for developing an implicit-explicit variant [2, 17].

The CFL condition for the explicit advection scheme is comparable with CFL conditions found in [8] for a number of explicit Runge-Kutta methods. Also the results reported in [3], where (3.13) is applied to a global, spherical advection problem, show that the explicit two-step integration formula combined with the $\kappa$-scheme works well. For the implicit-explicit method the CFL condition is certainly acceptable, since in practice stepsizes taken by this method will be determined mainly by the stiff chemistry and vertical diffusion and mostly be smaller than the largest permissible advection stepsize. Of course, the explicit advection approach should not reduce the excellent stability of the implicit BDF method for the stiff chemistry and vertical diffusion computation. The question thus is, will the implicit-explicit method be stable for any stepsize $\tau_a$ for which the explicit advection scheme is stable. This implies that the root condition for (3.10) must be satisfied for all possible values of $\tau_a A_\lambda$ and $\zeta = 1 - \frac{2}{3}\tau_a(A_\sigma + \mu)$. For $\zeta < 0$ this is true and a consequence of the theorem below. Note that we have $\zeta < 0$ if the chemistry eigenvalue $\mu$ is negative. It can be shown that the result is not true for all complex $\mu$ with $Re(\mu) \leq 0$. This, however, seems to be redundant for atmospheric chemical kinetics problems.

**Theorem 1** *Let the complex number $z \in S$. The roots of*

$$(1 - \frac{2}{3}\zeta)\alpha^2 - (\frac{4}{3} + \frac{4}{3}z)\alpha + (\frac{1}{3} + \frac{2}{3}z) = 0,$$

*then lie on the unit disk for any* $\zeta < 0$.

Proof. Let $\alpha_1, \alpha_2$ be the roots for $\zeta = 0$. As $z \in S$, we have $|\alpha_j| \leq 1, j = 1, 2$. Now write the characteristic equation as

$$(1 - \frac{\alpha_1}{\alpha})(1 - \frac{\alpha_2}{\alpha}) = \frac{2}{3}\zeta \tag{3.17}$$

and consider the stability domain for $\zeta$, for any fixed $z \in S$. On the boundary of this domain we have a root $|\alpha| = 1$, which implies that $|\alpha_j/\alpha| \leq 1$. It then follows from (3.17) that this boundary cannot intersect the negative axis and hence the entire negative $\zeta$-axis must belong to the stability domain since we have stability for $\zeta \to -\infty$. □

## 4. The performance for 3D applications

In this section we present test results of the complete implicit-explicit MOL scheme obtained for a concrete 3D test problem. We also discuss the speedup obtained by vectorization and parallelization (on a Cray C98/4256).

### *4.1 The 3D test problem*

The 3D test problem (1.1)-(1.3) is based on an extension of the 1D column model to a 3D model on the sphere. The cell-centered horizontal grid covers an area of $7.5^o$ square, arbitrarily chosen near the equator. This corresponds with 850 km square, approximately. We take a uniform longitude-latitude grid in the horizontal directions

$$\Omega_H = \{(\lambda_i, \phi_j) : \lambda_0 = -97.5^o,\ \phi_0 = -7.5^o,\ \lambda_i = (i - \tfrac{1}{2})\Delta,\ \phi_j = (j - \tfrac{1}{2})\Delta\ ;\ i, j = 1, \cdots, N\}, \tag{4.1}$$

with cell width $\Delta = 7.5^o/N$. In the vertical direction the domain definition and the discretization are the same as in the column model described in Section 2.5. Note that the emission input, deposition, etc. are also defined in the same way as in this column model. Thus we take the 'rural case' emission input on the whole domain, except for a square inside (see Figure 5) where we switch to the 'urban case' emission input (cf. [19]) which is approximately a factor 10 larger. This serves to create significantly larger (also an order of magnitude of 10) concentrations at the urban area, which then downwind gradually must be reduced to the lower rural levels. The discretization of the horizontal transport term (1.2) is based on the flux-limited finite difference scheme referred to in the beginning of Section 3. We use zero Neumann boundary conditions. A divergence-free windfield is imposed with such a direction and strength that the area is crossed diagonally in four days

$$u(\lambda, \phi) = \frac{\pi\alpha}{24}\frac{\sqrt{2}}{345600} \cdot (\cos\beta\cos\phi + \sin\beta\sin\phi\cos\lambda), \tag{4.2}$$

$$v(\lambda, \phi) = -\frac{\pi\alpha}{24}\frac{\sqrt{2}}{345600} \cdot \sin\beta\sin\lambda, \tag{4.3}$$

where $\beta = -3\pi/4$ is the angle with the equator. This corresponds with a wind speed of ca. 3.6 m/sec.

We take $N = 32, 64, 128$ and we start our computations at $t_0 = 14400 + 7200 \cdot 8$. As initial solution we take in all grid points the vertical 1D solution of the 'rural case' at that time. In our tests different resolutions were chosen to show the performance of the numerical algorithms in 3D models of regional to urban scale. Notice that the chosen values for $N$ correspond with gridsizes in the horizontal domain of approximately 26.6, 13.3, 6.6 km.

Let us elaborate the CFL restriction (3.16) for this test problem. For this purpose we first consider the corner point $(-90^o, 0^o)$, where

$$u = v = \frac{-\pi\alpha}{24 \cdot 345600}.$$

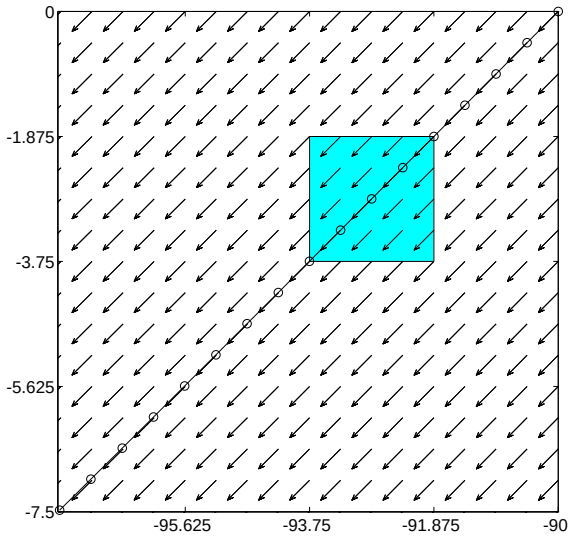


Figure 5: The horizontal domain with the windfield. In the gray square we impose 'urban' emissions, in the rest of the domain 'rural' emissions. The diagonal line gives the points where the solution is compared to the 1D solution along the characteristic.

Because $\phi = 0^o$, we have $\cos\phi = 1$ in (1.2) so that (3.16) becomes

$$\frac{\tau|u|}{\alpha\Delta} \leq 0.225.$$

Inserting $\Delta = 7.5^o/N = 2\pi/(48N)$ radians, this yields

$$\tau \leq 0.225 \cdot \frac{24 \cdot 345600}{\pi\alpha} \frac{\alpha\pi}{24N} \approx \begin{cases} 2430.0sec., & \text{for } N = 32, \\ 1215.0sec., & \text{for } N = 64, \\ 607.5sec., & \text{for } N = 128. \end{cases} \tag{4.4}$$

Because the angles $\phi, \lambda$ do not vary much over the space domain, these inequalities are approximately true everywhere. The linear stability analysis of Section 3.2 predicts that if (4.4) is satisfied, then the implicit-explicit BDF scheme will be stable. In the actual application variable stepsizes are used. Consequently, if violation of (4.4) would result in instability, then the local error control must detect the onset of it and reduce the stepsize to a stable level.

*4.2 Vectorization and parallelization*

For $N = 128$ the dimension of the complete semi-discrete ODE system equals $129 \cdot 129 \cdot 40 \cdot 66 \approx 44 \cdot 10^6$. Obviously, speed is then of utmost importance and because we use a computer with four vector processors (Cray C98/4256), a natural question is how to obtain good vector speed.

The integration scheme (3.3) has been implemented in a modular way. Separate routines perform the flux computations and the explicit advection part. A straightforward implementation of these is automatically optimized by the compiler both with respect to parallelization and to vectorization, resulting for these parts in a good performance. The subroutines are analogous to the ones used in [3].

The core of the implicit solver for the chemical transformations and the vertical turbulent diffusion is the Gauss-Seidel solver (2.13). This solver has to be applied for all horizontal grid points but the

computations are independent. In principle a loop over all horizontal grid points could be placed around (2.13), but the (current) compiler is not capable of optimizing this code by loop splitting and loop nest restructuring. Therefore we implemented loops over all horizontal grid points inside the items 2a-2d of (2.13). It was even necessary to collapse these loops by hand, since a double loop over the $\lambda$ and $\phi$ coordinates was not in all cases automatically collapsed. With this implementation the Gauss-Seidel process vectorizes very well. To obtain a good performance on a shared memory system with only a few processors item 2a can be parallelized over the vertical grid, while in items 2b and 2c the (collapsed) loop over the horizontal grid is distributed over the processors.

*4.3 Test results*

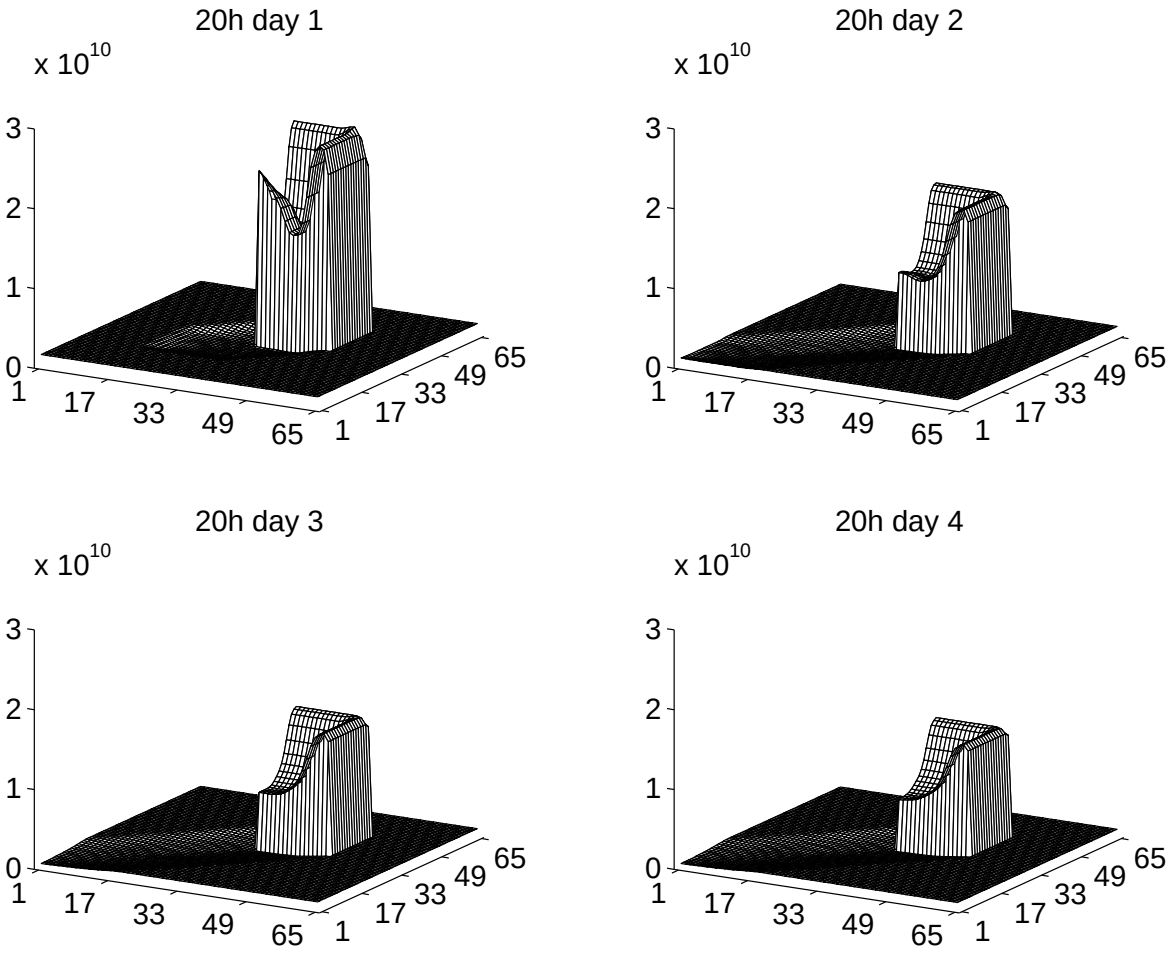


Figure 6: The $NO$ concentration at the end of each day on the horizontal domain ($N = 64$) in the first vertical layer.

In Figure 6 the horizontal distribution of the $NO$ concentration in the first vertical layer is plotted. It shows that the transitions between the rural and the urban area are very steep. Plots on the coarse grid ($N = 32$) show a comparable behavior. Likewise, Figure 7 shows the $O_3$ concentration. For $O_3$ the transitions between the rural and urban area are also steep, while downwind a strong dependence of the ozone level on the higher urban emission exists.

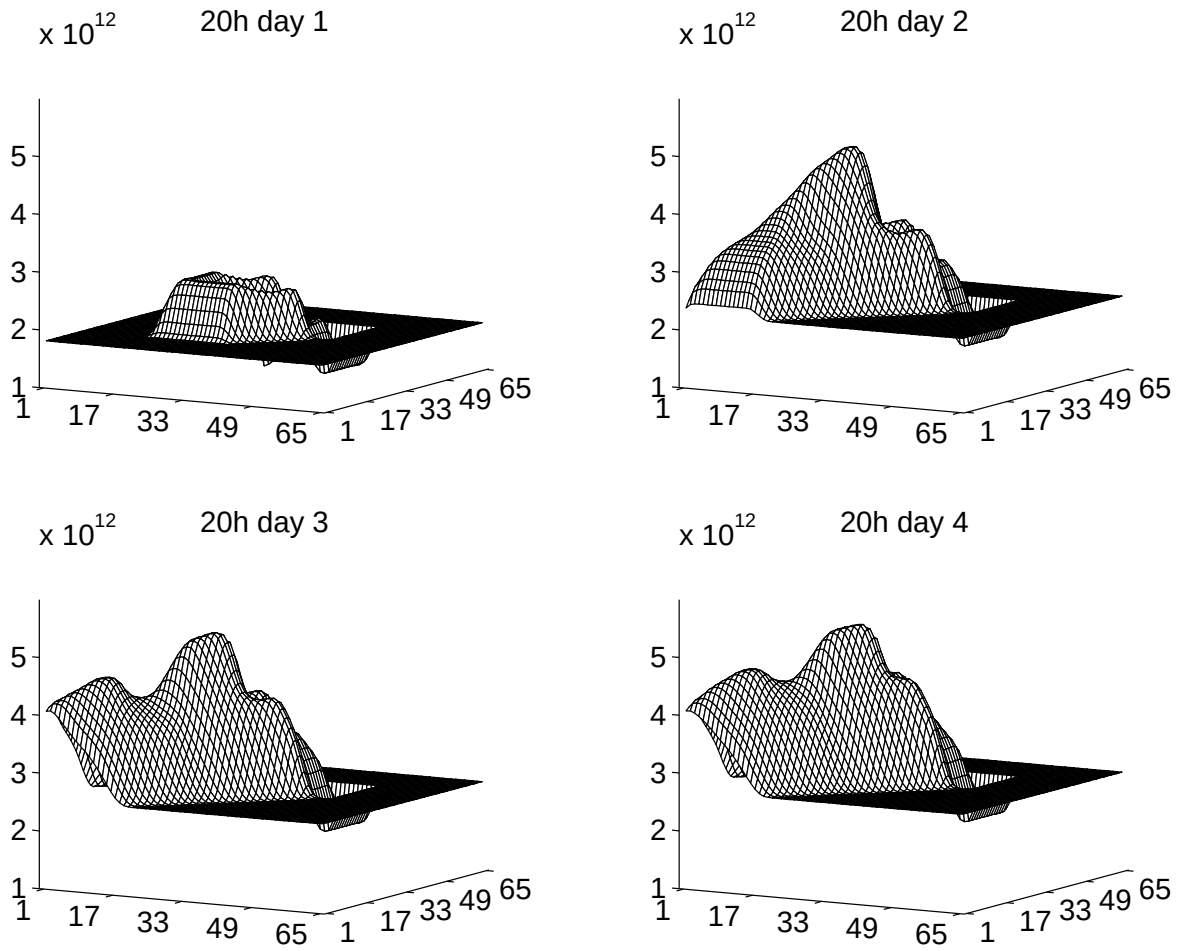


Figure 7: The ozone concentration at the end of each day on the horizontal domain ($N = 64$) in the first vertical layer.

We compare our numerical solution with the 1D solution obtained along the characteristic from $(-90^o, 0^o)$ to $(-97.5^o, -7.5^o)$. The error is measured by

$$ERR_j^n = \left[\sum_{k=1}^{40}(sol_{kj}^n - app_{kj}^n)^2\right]^{\frac{1}{2}} / \left[\max\left(1, \sum_{k=1}^{40}(sol_{kj}^n)^2\right)\right]^{\frac{1}{2}}, \tag{4.5}$$

$$SDA^n = -log_{10}\left(\frac{1}{66}\sum_{j=1}^{66} ERR_j^n\right), \tag{4.6}$$

$$SDA = -log_{10}\left(\frac{1}{N \cdot 66}\sum_{n=1}^{N}\sum_{j=1}^{66} ERR_j^n\right), \tag{4.7}$$

computed at all grid points through which the characteristic travels. Here, $sol_{kj}^n$ denotes a highly accurate approximation to the semi-discrete 1D solution along the characteristic at time $t_n = t_0 + n \cdot \frac{4 \cdot 24 \cdot 3600}{N}$ for $n = 1, \cdots, N$, which represents the true solution of the 3D problem at the vertical column

| | $129 \times 129 \times 40$ | | | $65 \times 65 \times 40$ | | | $33 \times 33 \times 40$ | | |
|---|---|---|---|---|---|---|---|---|---|
| RTol | SDA | CPUsec | Mflop/s | SDA | CPUsec | Mflop/s | SDA | CPUsec | Mflop/s |
| 0.01 | 1.71 | 4.45e+4 | 570 | 1.54 | 1.20e+4 | 550 | 1.24 | 3.31e+3 | 500 |
| 0.1 | 1.39 | 2.43e+4 | 570 | 1.44 | 6.06e+4 | 550 | 1.24 | 1.64e+3 | 500 |

Table 1: Performance on 1 CPU.

in the grid point $(N-n, N-n)$ at that time. Likewise, $app_{kj}^n$ denotes the numerical vertical column solution in this grid point at that time. (See Figure 5, where we indicated 17 of those points with a $\circ$.) Because we use variable stepsizes in time, linear interpolation in time is used to get an approximation at $t_n$.

We used four GS iterations and made runs for two different time tolerances, viz., $RTol = 0.1$ and $RTol = 0.01$. Like in the 1D case, after every two hours the integration is restarted with the backward Euler formula and a tenfold smaller stepsize. This of course somewhat enlarges the integration costs and is in fact redundant since we use a given constant wind field and keep also all meteorological parameters in the chemistry scheme fixed. In practice, however, we have to reckon with changes in input data at regular two hour intervals, say, motivating our use of restarts. At the $65 \times 65 \times 40$ grid, for $RTol = 0.1$ the maximum and average stepsize taken are, respectively, 2400 sec. and 307 sec. For $RTol = 0.01$ these values are 1118 sec. and 144. So this amounts to a total of 1125 integration steps for $RTol = 0.1$ and 2407 for $RTol = 0.01$.

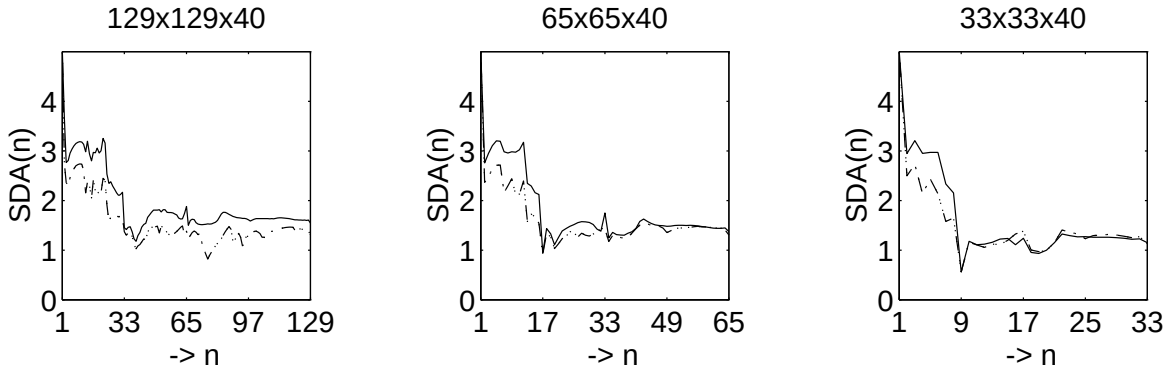


Figure 8: The errors at the grid points through which the characteristic travels, shown for the three different space grids and the two time tolerances $RTol = 0.1$ (dashdotted) and $RTol = 0.01$ (solid)

Table 1 shows that on the coarse grid, and to a lesser extent also for $N = 64$, $RTol = 0.01$ does not result in a more accurate solution. This obviously indicates that the spatial errors dominate. To illustrate this we have plotted in Figure 8 the $SDA^n$ values in all grid points through which the characteristic travels. Note that in the first quarter no spatial errors are present since in the right upper square of the domain the solution is constant over the horizontal grid. It is clear that the steep gradients in the rural-urban transition result in dips in the accuracy. On the coarse grids after the first transition the error is no longer influenced by the time tolerance and the spatial errors start to dominate. On the finest grid, $N = 128$, the accuracy is for $RTol = 0.1$ even lower than on the grid with $N = 64$. It can be seen in Figure 8 that the drop in accuracy occurs after the urban area. The reason is that we did not impose the CFL restriction (4.4) on the time stepsize assuming that the local error control would detect instabilities timely and reduce the stepsize to a stable level. This obviously did not happen during the nights when two time steps were taken of approximately $900sec.$ resulting in a drop of accuracy caused by instabilities in the area just SW of the urban square. A rerun starting

| | $129 \times 129 \times 40$ | | $65 \times 65 \times 40$ | | $33 \times 33 \times 40$ | |
|---|---|---|---|---|---|---|
| RTol=0.01 | CPUsec | Mflop/s | CPUsec | Mflop/s | CPUsec | Mflop/s |
| GS | 2.49e+4 | 610 | 6.50e+3 | 600 | 1.67e+3 | 590 |
| $\text{Flux}_\lambda$ | 7.00e+3 | 510 | 1.91e+3 | 500 | 6.53e+2 | 360 |
| $\text{Flux}_\phi$ | 6.58e+3 | 540 | 1.78e+3 | 530 | 4.80e+2 | 490 |
| | $129 \times 129 \times 40$ | | $65 \times 65 \times 40$ | | $33 \times 33 \times 40$ | |
| RTol=0.1 | CPUsec | Mflop/s | CPUsec | Mflop/s | CPUsec | Mflop/s |
| GS | 1.36e+4 | 610 | 3.29e+3 | 590 | 8.27e+2 | 590 |
| $\text{Flux}_\lambda$ | 3.83e+3 | 510 | 9.66e+2 | 500 | 3.24e+2 | 360 |
| $\text{Flux}_\phi$ | 3.60e+3 | 540 | 8.99e+2 | 530 | 2.38e+2 | 490 |

Table 2: Performance on 1 CPU for GS process and flux computations.

| | Fr.Par. | 4 | 8 | 16 |
|---|---|---|---|---|
| Overall | 97% | 3.4 | 5.4 | 7.7 |
| GS | 100% | 3.4 | 5.4 | 7.8 |
| $\text{Flux}_\lambda$ | 96% | 3.5 | 6.3 | 8.9 |
| $\text{Flux}_\phi$ | 96% | 3.5 | 6.1 | 8.7 |

Table 3: ATExpert data of the parallel fraction and the predictions of the performance on a dedicated machine for the whole program and for the GS process and the flux computations.

at $20h$ day 2 with the CFL restriction imposed confirmed this. Note that also for the more strict time tolerance $RTol = 0.01$ this accuracy drop caused by instabilities does not occur.

The space discretization scheme in the advection part will be first order at the rapid transition points due to the limiting procedure. Figure 8 confirms this, approximately. Beyond the first transition, the errors plotted in Figure 8 are close to 2% ($N = 128$), 4% ($N = 64$), and 6% ($N = 33$). These errors seem rather large but are believed to be quite acceptable for modeling purposes.

The vectorization of the code is very satisfying. To value the figures in Table 1 and 2, bear in mind that one CPU of a C90 has a clock period of 4.2ns and a double vector pipe. This gives a theoretical peak performance on 1 processor of 476 Mflop/s and 952 when chaining an add and a multiply. As was already shown in [3], the explicit part of the solver, which consists mainly of flux computations, has a performance of approximately 0.5 Gflop/s. The implicit part of the solver is dominated by the Gauss-Seidel process which reaches even 0.6 Gflop/s. To measure the Megaflop rate and the CPU time of a routine we used the Cray utility Perftrace[4], that gives the hardware performance by program unit.

Parallelization is done with the Cray Autotasking system which automatically distributes loop iterations to multiple processors, optionally guided by user directives. In case of Autotasking the Cray tool ATExpert[4] can be used to predict speedups for a number of processors on a dedicated system from data collected from a run on a nondedicated system. Table 3 shows the ATExpert information obtained on the $65 \times 65 \times 40$ grid. The parallel fraction gives an indication of the optimal speedup according to Amdahl's law $S = 1/(f_S + f_P/N)$, where $f_S$ and $f_P$ are the sequential and parallel fraction, respectively, and $N$ the number of processors. E.g., a parallel fraction of 97% gives a speedup of 3.7 on a 4 processor machine and 11.0 on 16 processors. The figures in the table indicate that the actual speedup would be much lower, especially for the GS process, where the optimal speedup of 16 on 16 processors is predicted to reduce to an actual speedup of only 7.8. There are mainly two reasons for this. The first is load imbalance. E.g., the computation of the production and loss terms in the Gauss-Seidel process is parallelized over the vertical grid (40 grid points) and gives an almost optimal speedup for 8 processors, but not for 16 processors. The second cause for a lower actual

speedup is the system time needed to invoke and terminate a parallel region (3600 clock periods on a C90). If the parallel sections do not contain enough work, most of the work will be done by the master task and a few slaves. On a $65 \times 65$ grid the actual speedup does not approach the optimal because the amount of work needed to do a part of the decomposition or the backsolve is in the order of a few chimes per loop iteration and because the loop over the horizontal grid needs to be parallelized as well as vectorized, resulting in a relatively small number of points per processor. On the $129 \times 129$ horizontal grid this overhead is less important and indeed an inspection of a few loops shows that the speedup is significantly larger than for the coarser grid.

## 5. Final remarks

As regards the temporal discretization, air pollution codes usually employ operator splitting. Following the method of lines, in this paper we have discussed a direct numerical integration based on the variable stepsize, 2nd order BDF formula. This choice of integration formula is a natural one in view of its excellent performance for stiff ODE problems from chemical kinetics. However, since we deal with a huge system of ODEs obtained after the spatial discretization of the advection and diffusion terms, it would be very cumbersome to apply the BDF formula in its fully implicit form. We therefore have modified it to an implicit-explicit form which treats advection explicitly and vertical turbulent diffusion and chemistry implicitly. As outlined in Section 3, this implicit-explicit modification is appropriate for our application as regards local accuracy and stability. As far as stability is concerned, we owe this to the 3rd-order upwind biased discretization of the advection operator.

The implicit-explicit modification means that the main processes of advection, chemical transformation and vertical turbulent diffusion are treated according to their general physical time constants. Advection is rather slow and can thus be treated explicitly. Certain chemical species and vertical turbulent diffusion have small time constants of the same magnitude. The error introduced in treating these two latter processes decoupled, as in operator splitting methods, is therefore difficult to estimate and can be avoided by solving them coupled and by an implicit approach.

However, the remaining 1D solution of the vertical turbulent diffusion and chemistry is a huge task, since this has to be done at every point from the horizontal grid. For this task we have developed the Gauss-Seidel technique (2.13), which for the 1D example problem from Section 2.4 has been shown to be 4 to 5 times more efficient than the usual modified Newton iteration supplied with a linear band solver. A second advantage of the Gauss-Seidel technique, compared to modified Newton, is its low memory requirement. No Jacobian matrices need to be stored which makes it possible to exploit grid vectorization in core memory for the very large problem sizes shown here. In view of the CPU times required, it is obvious that good vectorization and parallelization is a practical necessity. The flop rates of 0.5 Gflop/s, or half the peak performance, that we measured for the 3D test problem, illustrate that the vectorization of our implementation on the C90 is very satisfying. The algorithms used are naturally parallel. Without much effort the parallel fraction of the total program is 97%. On the $65 \times 65$ horizontal grid the actual speedup obtained by parallelization on 4 processors is 3.4 which is quite acceptable, but for a larger number of processors, say 16, the amount of work in some of the parallelized loops is too small to reach the optimal speedup of 11.0.

Our 3D test problem is realistic as regards the chemistry scheme. We used the EMEP MSC-W ozone chemistry which is state-of-the-art in the field of regional air pollution modeling. Atmospheric and meteorological conditions, like the vertical turbulent diffusion, the wind field, humidity and temperature, have been prescribed in analytic form and hence we have not simulated a genuine atmospheric pollution problem in all respects. However, numerically the example problem provides a challenging test. The tenfold higher urban emissions in part of the computational modeling domain give rise to sharp transition regions which are difficult for advection schemes. This, combined with the realistic chemistry and the vertical turbulent diffusion modeling, makes it a challenging 3D test problem well suited for benchmarking numerical codes, both with respect to the discretization aspects of accuracy

and stability and the high performance computing aspects of vectorization and parallelization.

ACKNOWLEDGEMENT
Discussions with David Simpson (Norwegian Meteorological Institute) on various aspects of the 1D test model are gratefully acknowledged. We thank Christoph Kessler for his comments on the final draft of the report.

REFERENCES

1. D.J. Allen, A.R. Douglass, R.B. Rood, and P.D. Guthrie. Application of a monotonic upstream-biased transport scheme to three-dimensional constituent transport calculations. *Monthly Weather Review*, 119:2456–2464, 1991.
2. U.M. Ascher, S.J. Ruuth, and B. Wetton. Implicit-explicit methods for time-dependent PDEs. Technical Report 93-15, Dept. of Comp. Science, University of British Columbia, 1993.
3. J.G. Blom, W. Hundsdorfer, and J.G. Verwer. Vectorization aspects of a spherical advection scheme on a reduced grid. Report NM-R9418, CWI, Amsterdam, 1994.
4. Cray Research, Inc. *UNICOS Performance Utilities Reference Manual*, SR-2040 6.0 edition, 1991.
5. J.J. Dongarra, C.B. Moler, J.R. Bunch, and G.W. Stewart. *LINPACK Users' Guide*. SIAM, Philadelphia, 1979.
6. E.Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer Verlag, New York, 1991.
7. Ø. Hov. Numerical solution of a simplified form of the diffusion equation for chemically reactive atmospheric species. *Atmospheric Environment*, 17:551–562, 1983.
8. W. Hundsdorfer, B. Koren, M. van Loon, and J.G. Verwer. A positive finite-difference advection scheme. *J. Comput. Phys.*, 116:xxx–xxx, 1995. Revision of CWI Report NM-R9309.
9. M.Z. Jacobson and R.P. Turco. SMVGEAR: A sparse-matrix, vectorized Gear code for atmospheric models. *Atmospheric Environment*, 28:273–284, 1994.
10. Ch. Kessler. *Entwicklung eines effizienten Lösungsverfahrens zur modellmäßigen Beschreibung der Ausbreitung und chemischen Umwandlung reaktiver Luftschadstoffe*. PhD thesis, Universität Karlsruhe, Germany, 1994.
11. O. Knoth and R. Wolke. Numerical methods for the solution of large kinetic systems. Report, Institut für Troposphärenforschung, Permoserstrasze 15, 04303 Leipzig, Germany, 1994.
12. G.J. McRae, W. R. Goodin, and J.H. Seinfeld. Numerical solution of the atmospheric diffusion equation for chemically reacting flows. *J. Comput. Phys.*, 45:1–42, 1982.
13. N. Moussiopoulos, editor. *The EUMAC Zooming Model: Model Structure and Applications*. EUROTRAC Report, EUROTRAC International Scientific Secretariat, Garmisch-Partenkirchen, Germany, 1994.
14. D. Simpson. Photochemical model calculations over Europe for two extended summer periods: 1985 and 1989. model results and comparisons with observations. *Atmospheric Environment*, 27A:921–943, 1993.
15. D. Simpson, Y. Andersson-Skold, and M.E. Jenkin. Updating the chemical scheme for the EMEP MSC-W model: Current status. Report EMEP MSC-W Note 2/93, The Norwegian Meteorological Institute, Oslo, 1993.
16. B.P. Sommeijer and J. Kok. Implementation and performance of a three-dimensional numerical transport model. Report NM-R9402, CWI, Amsterdam, 1994.

17. J.M. Varah. Stability restrictions on second order, three-level finite-difference schemes for parabolic equations. *SIAM J. Numer. Anal.*, 17:300–309, 1980.

18. J.G. Verwer. Gauss-Seidel iteration for stiff ODEs from chemical kinetics. *SIAM J. Sci. Comput.*, 15:1243–1250, 1994.

19. J.G. Verwer and D. Simpson. Explicit methods for stiff ODEs from atmospheric chemistry. Report NM-R9409, CWI, Amsterdam, 1994.

20. D.L. Williamson. Review of numerical approaches for modeling global transport. In H. van Dop and G. Kallos, editors, *Air Pollution Modeling and Its Application IX*, pages 377–394. Plenum Press, New York, 1992.