



Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

A Numerical Study for Global Atmospheric Transport-Chemistry
Problems

E.J. Spee, J.G. Verwer, P.M. de Zeeuw, J.G. Blom, W. Hundsdorfer

Modelling, Analysis and Simulation (MAS)

MAS-R9702 February 28, 1997

Report MAS-R9702
ISSN 1386-3703

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

CWI is the National Research Institute for Mathematics and Computer Science. CWI is part of the Stichting Mathematisch Centrum (SMC), the Dutch foundation for promotion of mathematics and computer science and their applications.

SMC is sponsored by the Netherlands Organization for Scientific Research (NWO). CWI is a member of ERCIM, the European Research Consortium for Informatics and Mathematics.

Copyright © Stichting Mathematisch Centrum
P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

A Numerical Study for Global Atmospheric Transport-Chemistry Problems

E.J. Spee, J.G. Verwer,
P.M. de Zeeuw, J.G. Blom, W. Hundsdorfer

CWI

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

Edwin.Spee@cwi.nl, Jan.Verwer@cwi.nl,
Paul.de.Zeeuw@cwi.nl, Joke.Blom@cwi.nl, W.Hundsdorfer@cwi.nl

ABSTRACT

Atmospheric air quality modeling relies in part on numerical simulation. Required numerical simulations are often hampered by lack of computer capacity and computational speed. This problem is most severe in the field of global modeling where transport and exchange of trace constituents are studied in the whole of the global troposphere/stratosphere. Studies in this field easily lead to computations with millions of unknowns over long time spans. Use of fast, tailored numerical algorithms implemented on the most advanced computer systems is then needed for making real progress. This paper is devoted to a performance study of algorithms developed for solving global atmospheric transport-chemistry problems. The algorithms are applied within two different operator splittings and are vectorized and parallelized on a Cray C90. For the performance study we use a constructed, three-space dimensional model problem, containing advection, vertical turbulent diffusion and chemical reactions. This numerical model problem is presented as a benchmark on which other algorithms and implementations can be tested. The benchmark problem together with all the algorithm software is available through World Wide Web. In the context of operator splitting, particular attention is paid to the issue of solving chemistry and vertical turbulent diffusion coupled or uncoupled.

1991 Mathematics Subject Classification: 65M06, 65M20, 65Y05, 65Y20

1991 Computing Reviews Classification System: G.1.7, G.1.8, J.2

Keywords and Phrases: Air quality modeling, global atmospheric transport-chemistry problems, operator splitting, advection schemes, stiff ODE solution, high performance computing, vectorization, parallelization.

Note: Work was carried out under project CIRK and was supported by Cray Research, Inc., under Grant 96.03, via the Stichting Nationale Computerfaciliteiten (National Computing Facilities Foundation, NCF) and by the National Institute of Public Health and Environmental Protection (RIVM).

1. INTRODUCTION

Air quality models are used to enhance the understanding of the chemical composition of the atmosphere, in particular with regard to the relation between emissions and the resulting distributions of primary and secondary polluting species. These models are mass balances which appear in the form of systems of time-dependent, three-space dimensional, partial differential equations (PDEs). These systems split additively into various subsystems describing advective transport, turbulent/diffusive transport, chemical transformations, emissions and depositions. Research into these models is becoming more and more important as it now seems evident that human activities leading to atmospheric air pollution can entail the danger of a long lasting global environmental change. The recent review papers [2, 12] discuss the current state and future directions in air quality modeling.

Atmospheric air quality modeling relies in part on numerical simulation as the PDEs used are nonlinear and cannot be solved by analytical means. However, numerical simulations are often severely

hampered by lack of computer capacity and computational speed. This problem is most severe in the field of global modeling where transport and exchange of trace constituents are studied in the whole of the global troposphere/stratosphere. Studies in this field easily lead to computations with millions of unknowns over long time spans. In such cases use of the most advanced computer systems is a prerequisite for making real progress in modeling research.

Bearing the practical problem of computational speed in mind, this paper is devoted to a performance study of algorithms tailored for solving global atmospheric transport-chemistry problems. The algorithms are applied and compared within two different operator splittings and are vectorized and parallelized on a Cray C90. Performance results are presented with respect to (grid) vectorization and parallelization based on autotasking combined with a division into subdomains of the globe.

For the performance study we have used a constructed, three-space dimensional model problem containing advection, vertical turbulent diffusion and a photochemical scheme consisting of 45 reactions between 17 species. This particular model problem has allowed us to examine the numerical accuracy/efficiency of the algorithms and the operator splittings. In the context of operator splitting, particular attention is paid to the issue of solving chemistry and vertical turbulent diffusion coupled or uncoupled. The model is restricted in the sense that it does not simulate a real atmosphere. However, for numerical testing purposes it is valuable, certainly with regard to the chemistry which is based on a set of photochemical reactions from practice. The model is therefore presented as a benchmark on which other schemes and implementations can be tested as well, since we consider benchmarking important for algorithm and code development.

The paper is organized as follows. In Section 2 we discuss the spherical mass balance equation used in our investigation. Section 3 is devoted to the numerical algorithms and the two operator splittings. The benchmark problem is discussed in Section 4. Section 5 presents the results on vectorization, showing Mflop rates and CPU times, and discusses accuracy and efficiency. Section 6 is devoted to the issue of parallelization. The final Section 7 summarizes conclusions and remarks.

2. THE TRANSPORT-CHEMISTRY MODEL

At the heart of atmospheric air quality models lie mass balances in the form of systems of advection-diffusion-reaction equations

$$\frac{\partial c}{\partial t} + \nabla \cdot (\underline{u}c) = \nabla \cdot \left(\rho D \nabla \left(\frac{c}{\rho} \right) \right) + R(c). \quad (2.1)$$

The unknown c denotes a vector of species concentrations, say of length m , and ρ is the density of the air. The velocity wind field vector \underline{u} and the diffusion matrix D are supposed to be known (off-line model). The wind field is generated separately by circulation models or retrieved from data bases and the diffusion is usually determined by atmospheric turbulence models. While advection and diffusion determine the transport of species, the reaction term R represents the atmospheric chemical reactions, depositions and emissions. Note that the reaction term couples all equations in the model. All processes are time and space dependent, but this dependence is suppressed in our notation.

The vector function $R(c)$ has the special form

$$R(c) = P(c) - L(c)c, \quad (2.2)$$

where $P(c)$ is the vector of production terms and $L(c)c$ the vector of loss terms with $L(c)$ a diagonal matrix. For many species, the reciprocal of their entry in $L(c)$ is a good approximation of the physical time constant or characteristic reaction time. In virtually all applications, the range of reaction times is huge, ranging from milliseconds or shorter (e.g. OH radical) to years (e.g. CH_4). This means that atmospheric chemistry gives rise to stiffness. A second important feature is photochemistry, giving rise to rapid changes in concentration values at sunrise and sunset. We thus have to face constantly moving areas of rapid solution change. In applications the number of species varies. Current global models use about 20 species, but in [2, 12] it is pointed out that as many as 40 to 100 species are necessary for an adequate analysis of perturbations to atmospheric chemistry on a global and regional

scale. These review papers also point out that grid resolutions of $1^\circ \times 1^\circ$ or better in the horizontal and at least 20 vertical grid layers are needed to achieve this. Altogether this means that global and regional air quality modeling leads to a huge computational task. Even with high-performance computers at hand, computer capacity often dictates the grid resolution.

As far as transport is concerned, in this paper we restrict ourselves to horizontal transport by wind and vertical transport by turbulent diffusion. This means that vertical advection and horizontal diffusion is neglected. In reality these two processes are of less importance and adding them at a later stage will not lead to large numerical difficulties. A greater restriction is that we here also neglect orography, which means that the earth is taken to be a real sphere. Nor do we use real meteorological data and hence are not confronted with massive I/O operations. In a sequel to this work, orography and the use of real meteorological data will be taken up.

We write $c = c(t, \lambda, \phi, r)$ where $\lambda \in [0, 2\pi]$ denotes longitude, $\phi \in [-\frac{\pi}{2}, +\frac{\pi}{2}]$ latitude and $r > 0$ the radial distance. With the above restrictions in mind, we put $\underline{u} = (u, v)$ and write the divergence in spherical coordinates,

$$\nabla \cdot (\underline{u}c) = \frac{1}{a \cos \phi} \left[\frac{\partial(u c)}{\partial \lambda} + \frac{\partial(v c \cos \phi)}{\partial \phi} \right]. \quad (2.3)$$

Here a is the radius of the earth (6378 km) and we should mention that we have approximated the radial distance factor $1/(r + a)$ by $1/a$. For the applications this is allowed, since the atmospheric layer (troposphere/stratosphere) in which models are used is extremely thin compared with the radius of the earth. The diffusion term becomes

$$\nabla \cdot \left(\rho D \nabla \left(\frac{c}{\rho} \right) \right) = \frac{\partial}{\partial r} \left(\rho K \frac{\partial}{\partial r} \left(\frac{c}{\rho} \right) \right), \quad (2.4)$$

where K is a scalar, vertical diffusion coefficient. We suppose for r the interval $0 \leq r \leq r_H$ with r_H a height uniform over the globe. As boundary conditions we use the no-flux conditions

$$\rho K \frac{\partial}{\partial r} \left(\frac{c}{\rho} \right) = 0, \quad r = 0, r_H. \quad (2.5)$$

Boundary conditions in the horizontal directions are not needed of course. The initial condition at the initial time $t = t_0$ is imposed by prescribing all species concentrations.

3. OPERATOR SPLITTING AND DISCRETIZATION

As outlined above, mathematical air quality models split additively into advective transport, turbulent/diffusive transport, chemical transformations, emissions and depositions. Virtually all computer implementations employ this in the numerical solution process by using the concept of operator (time) splitting. Splitting is popular because the submodels are different in nature and are therefore easier to solve apart than when kept together in the numerical treatment. However, splitting also has a number of disadvantages. First, it introduces an additional error, called the splitting error. This error exists even if submodels would be solved exactly, while in practice this error is hard to relate with the discretization errors. Second, splitting introduces strong initial transients for the fast reacting species in the chemistry computation and does this within each split time step. Due to the nonlinearity, it may be necessary to resolve these transients accurately in order to avoid instabilities and inaccuracies which cannot be corrected at later times. Resolving the transients accurately is costly, however. We thus see that it is of practical interest to apply operator splitting in a way that both these disadvantages are somehow reduced.

In this paper we compare two splitting procedures. The first one is standard, while in the second we avoid splitting the chemistry and vertical diffusion, applying the same numerical technique as in [22]. A related technique can be found in [9]. Keeping the chemistry and vertical diffusion coupled in the numerical solution process is natural, since both these processes contain small and large time scales (vertical turbulent diffusion transports species as fast as many react with one another). Comparative

results in [5] for a number of 1D diffusion-chemistry models clearly indicate that uncoupling chemistry and vertical turbulent diffusion can lead to substantial errors. We thus anticipate that in our 3D advection-diffusion-reaction case the second splitting procedure will perform notably better than the first one. In the remainder of this section we will first describe the two splitting procedures and after that we will present the algorithms for the submodels.

3.1 Operator splitting

We write equation (2.1) as

$$\frac{\partial}{\partial t}c = F_0(c) + F_1(c) + F_2(c), \quad (3.6)$$

where

$$F_0(c) = R(c), \quad F_1(c) = \frac{\partial}{\partial r} \left(\rho K \frac{\partial}{\partial r} \left(\frac{c}{\rho} \right) \right), \quad F_2(c) = -\frac{1}{a \cos \phi} \left[\frac{\partial(u c)}{\partial \lambda} + \frac{\partial(v c \cos \phi)}{\partial \phi} \right].$$

Let Δt_{split} denote the split step size and c_s the approximation to the exact concentration vector c , resulting from the splitting, at time $t = t_s$ where $t_s = s \Delta t_{\text{split}}$ for $s = 0, 1, \dots$. Starting from c_s , the approximation c_{s+1} at the next time point t_{s+1} is then computed with the standard splitting procedure by solving

$$\frac{\partial}{\partial t}c^{(1)}(t) = F_2(c^{(1)}(t)) \quad (t_s \leq t \leq t_{s+\frac{1}{2}}), \quad c^{(1)}(t_s) = c_s, \quad (3.7a)$$

$$\frac{\partial}{\partial t}c^{(2)}(t) = F_1(c^{(2)}(t)) \quad (t_s \leq t \leq t_{s+\frac{1}{2}}), \quad c^{(2)}(t_s) = c^{(1)}(t_{s+\frac{1}{2}}), \quad (3.7b)$$

$$\frac{\partial}{\partial t}c^{(3)}(t) = F_0(c^{(3)}(t)) \quad (t_s \leq t \leq t_{s+1}), \quad c^{(3)}(t_s) = c^{(2)}(t_{s+\frac{1}{2}}), \quad (3.7c)$$

$$\frac{\partial}{\partial t}c^{(4)}(t) = F_1(c^{(4)}(t)) \quad (t_{s+\frac{1}{2}} \leq t \leq t_{s+1}), \quad c^{(4)}(t_{s+\frac{1}{2}}) = c^{(3)}(t_{s+\frac{1}{2}}), \quad (3.7d)$$

$$\frac{\partial}{\partial t}c^{(5)}(t) = F_2(c^{(5)}(t)) \quad (t_{s+\frac{1}{2}} \leq t \leq t_{s+1}), \quad c^{(5)}(t_{s+\frac{1}{2}}) = c^{(4)}(t_{s+\frac{1}{2}}), \quad (3.7e)$$

and $c_{s+1} \equiv c^{(5)}(t_{s+1})$. By means of (3.7), the original problem thus has been splitted into five subproblems, of which three are different and can be treated with different numerical techniques. Note that the sequence of computations, in the order advection/vertical diffusion/chemistry/vertical diffusion/advection, is symmetrical around the time point halfway (Strang splitting [18]). This way of splitting is standard in the sense that all processes are treated uncoupled in a sequential manner. In the remainder, this standard splitting procedure will be called the type I splitting procedure. Problem (3.7c) will be referred to as the chemistry problem, problem (3.7a) and (3.7e) as the advection problem, and problem (3.7b) and (3.7d) as the diffusion problem.

The second splitting procedure, where the chemistry and vertical diffusion remain coupled, reads

$$\frac{\partial}{\partial t}c^{(1)}(t) = F_2(c^{(1)}(t)) \quad (t_s \leq t \leq t_{s+\frac{1}{2}}), \quad c^{(1)}(t_s) = c_s, \quad (3.8a)$$

$$\frac{\partial}{\partial t}c^{(2)}(t) = F_0(c^{(2)}(t)) + F_1(c^{(2)}(t)) \quad (t_s \leq t \leq t_{s+1}), \quad c^{(2)}(t_s) = c^{(1)}(t_{s+\frac{1}{2}}), \quad (3.8b)$$

$$\frac{\partial}{\partial t}c^{(3)}(t) = F_2(c^{(3)}(t)) \quad (t_{s+\frac{1}{2}} \leq t \leq t_{s+1}), \quad c^{(3)}(t_{s+\frac{1}{2}}) = c^{(2)}(t_{s+\frac{1}{2}}), \quad (3.8c)$$

and $c_{s+1} \equiv c^{(3)}(t_{s+1})$. This splitting procedure is nonstandard and will henceforth be called the type II splitting procedure. Problem (3.8b) will be referred to as the chemistry-diffusion problem.

In both splittings, the step size Δt_{split} must be chosen somehow in relation with the discretization errors made in the substeps. In general this choice is hard to make and requires experimental insight. Logical is to take the minimal value $\Delta t_{\text{split}} = 2 \Delta t_{\text{adv}}$, where Δt_{adv} denotes the step size for the advection computation. We have done this in all tests presented in this paper.

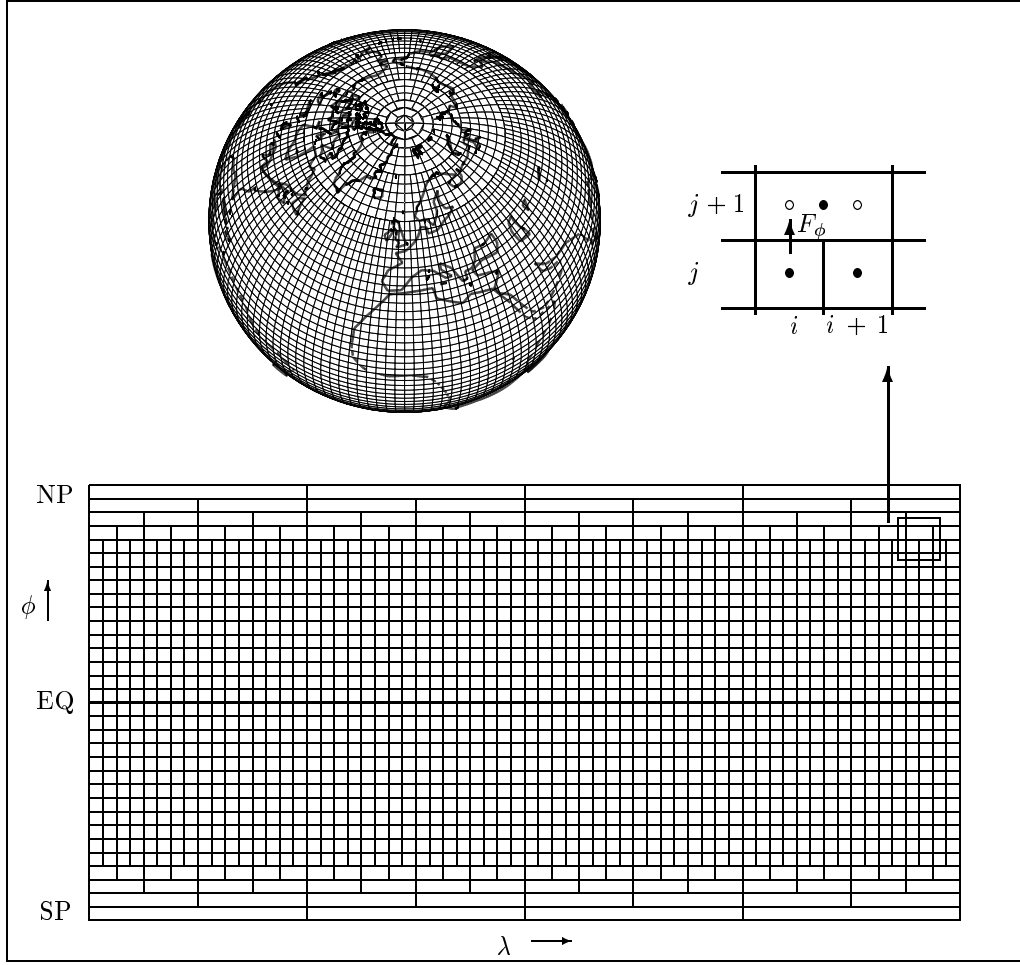


Figure 1: A reduced grid on the globe (left, top), virtual concentrations \circ and real concentrations \bullet (right, top), and a 64×32 reduced grid in longitude/latitude co-ordinates (bottom).

3.2 The advection computation

For both splittings the advection computation goes the same. Because there is only horizontal advection, it suffices to describe this computation for the 2D equation

$$\frac{\partial c}{\partial t} = - \frac{1}{a \cos \phi} \left[\frac{\partial(u c)}{\partial \lambda} + \frac{\partial(v c \cos \phi)}{\partial \phi} \right], \quad (3.9)$$

where $c = c(t, \lambda, \phi)$ is now a scalar rather than a vector. A so-called reduced space grid [24] is used, see Figure 1 for an example. Grid reduction means that at a small number of latitudes near the poles the grid size in the longitudinal direction is doubled. Without this doubling, a realistic grid on the globe would become very fine in the longitudinal direction near the poles, which imposes a severe stability restriction for explicit advection schemes.

The spherical advection operator is cell-centered discretized by a mass-conservative, flux-limited, third order upwind scheme. Let $c_{i,j}(t)$ denote the resulting approximation at a cell-center (λ_i, ϕ_j) with cell size $\Delta\lambda(j) \times \Delta\phi$ ($\Delta\lambda$ depends on the latitude index j due to the grid reduction). The semi-discrete counterpart of (3.9) then reads

$$\frac{d}{dt} c_{i,j} = -\frac{1}{a \cos \phi_j} \left[\frac{fL_{i+\frac{1}{2},j} - fL_{i-\frac{1}{2},j}}{\Delta \lambda(j)} + \frac{fP_{i,j+\frac{1}{2}} - fP_{i,j-\frac{1}{2}}}{\Delta \phi} \right], \quad (3.10)$$

with the fluxes on the cell boundaries

$$fL_{i+\frac{1}{2},j} = \max(u_{i+\frac{1}{2},j}, 0) fL_{i+\frac{1}{2},j}^+ + \min(u_{i+\frac{1}{2},j}, 0) fL_{i+\frac{1}{2},j}^-, \quad (3.11a)$$

$$fP_{i,j+\frac{1}{2}} = [\max(v_{i,j+\frac{1}{2}}, 0) fP_{i,j+\frac{1}{2}}^+ + \min(v_{i,j+\frac{1}{2}}, 0) fP_{i,j+\frac{1}{2}}^-] \cos(\phi_{j+\frac{1}{2}}). \quad (3.11b)$$

These fluxes are defined by

$$fL_{i+\frac{1}{2},j}^+ = c_{i,j} + \Psi(\theta_{i+\frac{1}{2},j}) \cdot (c_{i+1,j} - c_{i,j}), \quad (3.12a)$$

$$fL_{i+\frac{1}{2},j}^- = c_{i+1,j} + \Psi(\theta_{i+\frac{3}{2},j}^{-1}) \cdot (c_{i,j} - c_{i+1,j}), \quad (3.12b)$$

$$fP_{i,j+\frac{1}{2}}^+ = c_{i,j} + \Psi(\theta_{i,j+\frac{1}{2}}) \cdot (c_{i,j+1} - c_{i,j}), \quad (3.12c)$$

$$fP_{i,j+\frac{1}{2}}^- = c_{i,j+1} + \Psi(\theta_{i,j+\frac{3}{2}}^{-1}) \cdot (c_{i,j} - c_{i,j+1}), \quad (3.12d)$$

with

$$\theta_{i+\frac{1}{2},j} = \frac{c_{i,j} - c_{i-1,j}}{c_{i+1,j} - c_{i,j}}, \quad \theta_{i,j+\frac{1}{2}} = \frac{c_{i,j} - c_{i,j-1}}{c_{i,j+1} - c_{i,j}}, \quad (3.13)$$

and Ψ is the flux-limiter function [10]

$$\Psi(\theta) = \max(0, \min(1, \theta, \frac{1}{3} + \frac{1}{6}\theta)). \quad (3.14)$$

We used this advection approximation earlier in [1, 6, 15]. In [6] the particular type of flux-limiting is discussed. Flux-limiting serves to maintain positivity. If we switch off the flux-limiting, i.e. put $\Psi(\theta) = 1/3 + \theta/6$, the underlying third order upwind scheme is recovered.

Our reduced grid approach was developed in [1] where we refer to for a detailed discussion of the technicalities encountered. Only minor differences exist between the current implementation and the one developed in [1]. For example, at the particular circles of latitude where the grid reduction takes place, the flux fP on the coarse grid is defined as the sum of the corresponding fine grid fluxes in longitude direction, while in [1] the mean is used. Also, where grid reduction takes place, piecewise constant interpolation is used for concentration values whenever needed versus linear interpolation in [1]. Piecewise constant interpolation was found to work equally well.

At this stage of development, (3.10) is still time-continuous. For the time integration an explicit Runge-Kutta method is used, viz. the second order, two-stage explicit trapezoidal rule. Let us represent (3.10) by the system

$$\frac{d}{dt} \vec{c} = \vec{f}(\vec{c}), \quad (3.15)$$

where $\vec{c} = (c_{i,j})$ stands for the entire grid function for the concentrations and $\vec{f} = (f_{i,j})$ for the entire grid function formed by the right-hand sides of (3.10), here denoted by $f_{i,j}$. The complete advection scheme is then given by

$$\vec{w} = \vec{c}_m + \Delta t_{\text{adv}} \vec{f}(\vec{c}_m), \quad (3.16a)$$

$$\vec{c}_{m+1} = \vec{c}_m + \frac{1}{2} \Delta t_{\text{adv}} \left(\vec{f}(\vec{c}_m) + \vec{f}(\vec{w}) \right), \quad (3.16b)$$

where \vec{c}_m approximates $\vec{c}(t)$ at a time point $t = t_m$ and $\Delta t_{\text{adv}} = t_{m+1} - t_m$ is the advection step size. The time points t_m lie in a split interval $[t_s, t_{s+1}]$ as introduced previously. Usually, $\Delta t_{\text{adv}} = \Delta t_{\text{split}}/2$ or a smaller integer fraction. In all tests presented in this paper we have used $\Delta t_{\text{adv}} = \Delta t_{\text{split}}/2$.

The use of this second order two-stage Runge-Kutta method leads to two evaluations of the advection operator within each advection step. Noteworthy is that we have turned a linear advection problem into a nonlinear semi-discrete system (3.15) by applying flux-limiting. This leads to additional costs for the advection computation. However, in [6] it is shown that the combination, formed by the flux-limited third order discretization and this Runge-Kutta method, combines good stability with good positivity properties (see in particular Sections 3.3, 5.3 of [6]). The results presented there indicate that the combination is stable and positive, as long as Δt_{adv} is adjusted to satisfy the CFL restriction

$$\max_{i,j} (\nu_{\lambda_{i,j}} + \nu_{\phi_{i,j}}) \leq 2/3, \quad (3.17)$$

where

$$\nu_{\lambda_{i,j}} = \frac{\Delta t_{\text{adv}}}{\Delta \lambda} \frac{|u(\lambda_{i+\frac{1}{2}}, \phi_j)|}{a \cos \phi_j}, \quad \nu_{\phi_{i,j}} = \frac{\Delta t_{\text{adv}}}{\Delta \phi} \frac{|v(\lambda_i, \phi_{j+\frac{1}{2}})| \cos \phi_{j+\frac{1}{2}}}{a \cos \phi_j}. \quad (3.18)$$

3.3 The chemistry-diffusion computation used in type II splitting

Both the chemistry and diffusion computation used in type I splitting can be seen as simplified cases of the chemistry-diffusion computation used in type II splitting. We therefore first present the computation for the chemistry-diffusion problem

$$\frac{\partial}{\partial t} c = \frac{\partial}{\partial r} \left(\rho K \frac{\partial}{\partial r} \left(\frac{c}{\rho} \right) \right) + R(c), \quad 0 \leq r \leq r_H, \quad (3.19)$$

subjected to the boundary conditions (2.5), and recall that this computation has to be carried out over all split intervals $[t_s, t_{s+1}]$ at any of the cell centers (λ_i, ϕ_j) of the horizontal grid.

The usual numerical approach for such stiff problems leads us to implicit ODE solvers employing a form of Newton iteration for solving the implicit relations. However, despite the 1D nature, Newton iteration is still quite expensive here, as c is vector valued (m species). In [21, 22] an alternative technique has been proposed which for tropospheric models is significantly more efficient. This technique is based on a Gauss-Seidel iteration, which treats the chemistry in a scalarly implicit way and the diffusion implicitly. As a result, at most linear tridiagonal systems of algebraic equations need to be solved, rather than the much larger banded linear systems arising in the Newton process. We have adopted this technique for the current coupled chemistry-diffusion computation. For reasons of self-containedness, we here repeat the description of [22], Section 2.

It suffices to consider the 1D case so that $c = c(t, r) \in \mathbb{R}^m$. Let us introduce the mixing ratio vector $\chi(t, r) = c(t, r) / \rho(t, r)$. The diffusion term is discretized on the nonuniform cell-centered grid

$$\Omega_V = \{r_k : r_1 = \frac{1}{2}\Delta r_1, r_k = r_{k-1} + \frac{1}{2}(\Delta r_{k-1} + \Delta r_k), 2 \leq k \leq N_r\} \quad (3.20)$$

in the following way,

$$\frac{\partial}{\partial r} \left(\rho K \frac{\partial}{\partial r} \left(\frac{c}{\rho} \right) \right) \approx \frac{2}{r_{k+1} - r_{k-1}} \left((\rho K)_k^+ \frac{\chi_{k+1} - \chi_k}{r_{k+1} - r_k} - (\rho K)_k^- \frac{\chi_k - \chi_{k-1}}{r_k - r_{k-1}} \right), \quad (3.21)$$

for $1 \leq k \leq N_r$, with $\chi_k(t)$ denoting the semi-discrete approximation to $\chi(t, r_k)$ and

$$(\rho K)_k^\pm = (\rho K)(t, (r_k + r_{k\pm 1})/2), \quad r_0 = r_1 - \Delta r_1, \quad r_{N_r+1} = r_{N_r} + \Delta r_{N_r}.$$

Note that ρK is evaluated halfway between the cell centers, rather than at the cell boundaries, to obtain a discretization with order of consistency at least one on a nonuniform grid. The boundary conditions are incorporated by putting $(\rho K)_k^- = 0$ for $k = 1$ and $(\rho K)_k^+ = 0$ for $k = N_r$.

For the remainder it is important to recall that the diffusion operator introduces no coupling between different species. The species are coupled only through the chemistry system

$$R(c_k) = P(c_k) - L(c_k) c_k.$$

Let $c_k^{(j)}$ denote the j -th (component) species of c_k and introduce the following species vectors on Ω_V ,

$$\mathbf{c}^{(j)} = [c_1^{(j)}, \dots, c_{N_r}^{(j)}]^T, \quad \mathbf{P}^{(j)}(\mathbf{c}) = [P^{(j)}(c_1), \dots, P^{(j)}(c_{N_r})]^T, \quad j = 1, \dots, m. \quad (3.22)$$

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

$$\mathbf{L}^{(j)}(\mathbf{c}) = \text{diag}(L^{(j)}(c_1), \dots, L^{(j)}(c_{N_r})), \quad j = 1, \dots, m. \quad (3.23)$$

Then we may represent the semi-discrete ODE system resulting from the spatial discretization, by

$$\frac{d}{dt} \mathbf{c}^{(j)} = \mathbf{A} \mathbf{c}^{(j)} + \mathbf{P}^{(j)}(\mathbf{c}) - \mathbf{L}^{(j)}(\mathbf{c}) \mathbf{c}^{(j)}, \quad j = 1, \dots, m, \quad (3.24)$$

where \mathbf{A} is the tridiagonal diffusion matrix of order N_r .

This ODE system has to be integrated in time over each split interval $[t_s, t_{s+1}]$ introduced previously. At the beginning of each interval $[t_s, t_{s+1}]$, the integration is started with the well-known, first order, one-step implicit Euler rule. Thereafter the second order, two-step implicit backward differentiation (BDF) formula is used. This combination yields second order accurate time stepping which for atmospheric transport applications is sufficient in view of the modest accuracy requirement. Generally, a relative accuracy larger than 1% is superfluous. In our tests described later we have integrated with a variable step size within each split interval. The step size is hereby governed by a standard local error control mechanism similar as used in [20, 22]. Step sizes are taken variable so as to allow smaller step sizes in the initial part of the split intervals, where we may encounter rapid transient solution components connected with short living chemical species. However, we always impose a minimum for the step size (specified later), since we wish to avoid adjustment to the smallest time constants of radicals.

The two-step BDF formula is defined as follows. Let $\Delta t_{\text{cvd}} = t_{n+1} - t_n$ denote the step size, assuming that $t_n \in [t_s, t_{s+1}]$ is a time point for the BDF formula. Then we have

$$\mathbf{c}_{n+1}^{(j)} = \mathbf{C}^{(j)} + \gamma \Delta t_{\text{cvd}} \left(\mathbf{A} \mathbf{c}_{n+1}^{(j)} + \mathbf{P}^{(j)}(\mathbf{c}_{n+1}) - \mathbf{L}^{(j)}(\mathbf{c}_{n+1}) \mathbf{c}_{n+1}^{(j)} \right), \quad j = 1, \dots, m, \quad (3.25)$$

where the scalar γ and the vector $\mathbf{C}^{(j)}$ are BDF quantities defined by

$$\gamma = \frac{1+q}{1+2q}, \quad \mathbf{C}^{(j)} = \frac{1}{1+2q} \left((1+q)^2 \mathbf{c}_n^{(j)} - q^2 \mathbf{c}_{n-1}^{(j)} \right), \quad q = \frac{t_{n+1} - t_n}{t_n - t_{n-1}}. \quad (3.26)$$

If we put $\gamma = 1$ and $\mathbf{C}^{(j)} = \mathbf{c}_n^{(j)}$, the implicit Euler rule is recovered.

The BDF formula (3.25) is implicit and thus we need an iterative technique to approximately solve the $m \times N_r$ dimensional systems of nonlinear algebraic equations. As mentioned above, for this purpose we use a special Gauss-Seidel iteration which we describe next. For convenience of notation, we suppress the time index $n+1$ and put $\tau = \Delta t_{\text{cvd}}$, so that (3.25) now reads

$$\mathbf{c}^{(j)} = \mathbf{C}^{(j)} + \gamma \tau \mathbf{A} \mathbf{c}^{(j)} + \gamma \tau \mathbf{P}^{(j)}(\mathbf{c}) - \gamma \tau \mathbf{L}^{(j)}(\mathbf{c}) \mathbf{c}^{(j)}, \quad j = 1, \dots, m. \quad (3.27)$$

This system is equivalent to

$$\mathbf{c}^{(j)} = \left(\mathbf{I} - \gamma \tau \mathbf{A} + \gamma \tau \mathbf{L}^{(j)}(\mathbf{c}) \right)^{-1} \left(\mathbf{C}^{(j)} + \gamma \tau \mathbf{P}^{(j)}(\mathbf{c}) \right), \quad j = 1, \dots, m, \quad (3.28)$$

since the inverse of the tridiagonal matrix $\mathbf{I} - \gamma \tau \mathbf{A} + \gamma \tau \mathbf{L}^{(j)}(\mathbf{c})$ always exists. The Gauss-Seidel iteration for approximating $\mathbf{c}^{(j)}$, $1 \leq j \leq m$, is carried out on equation (3.28) and consists of the following calculations. Let $\mathbf{c}_{[i]}$ denote the i -th iterate for \mathbf{c} . Then, at integration step n , we have

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 $j = 1, \dots, m$:

- 2a. $\mathbf{L}^{(j)}(\mathbf{c}_{[i]}), \mathbf{P}^{(j)}(\mathbf{c}_{[i]}).$
- 2b. LU-decompose $\mathbf{I} - \gamma\tau\mathbf{A} + \gamma\tau\mathbf{L}^{(j)}(\mathbf{c}_{[i]}).$
- 2c. Backsolve $\mathbf{c}_{[i+1]}^{(j)}.$
- 2d. Update $\mathbf{c}_{[i]} = (\mathbf{c}_{[i+1]}^{(1)}, \dots, \mathbf{c}_{[i+1]}^{(j)}, \mathbf{c}_{[i]}^{(j+1)}, \dots, \mathbf{c}_{[i]}^{(m)}).$
3. Put $i := i + 1$. If more iterations are required, then go to 2.

Hence the approximations are corrected specieswise and simultaneously over the grid, such that the diffusion term is treated implicitly. This requires the tridiagonal matrix calculations 2b, 2c any time a species is corrected. No Jacobian matrices for the chemistry system are computed and no additional storage is required.

This particular Gauss-Seidel iteration is applied with a fixed number of iterations. Usually only a few iterations already lead to an efficient process. In the tests of this paper, we in fact use only 2 iterations throughout. Although in general we then do not expect to have come very close to the implicit BDF solution, our experience is that using only a few iterations leads to a stable and efficient solution process for tropospheric chemistry models. Comparisons in [21, 22] with the usual approach based on modified Newton iteration illustrate this. The number of iterations to choose is of course problem dependent. While in [21, 22] 4 iterations were advocated, a few trial and error runs showed that in the present case 2 iterations will do (see also Section 3.6).

3.4 The chemistry computation used in type I splitting

Would there be no vertical diffusion ($K = 0$), then the above chemistry-diffusion computation reduces to the chemistry computation used for the box models in [20, 23]. The same integration formulas are used, the Gauss-Seidel technique is the same and the method is also applied with variable step sizes in the same way as in the chemistry-diffusion computation. We can copy this box model process for the chemistry computation in our type I splitting, since in this splitting we only encounter box models, one at each spatial grid point. The step size for this process will be denoted by Δt_{che} instead of Δt_{cvd} .

3.5 The diffusion computation used in type I splitting

On the other hand, without chemistry, the chemistry-diffusion computation reduces to a standard tridiagonal, linearly implicit diffusion integration with the BDF2 formula, one for each vertical column and each species. The Gauss-Seidel process is then no longer operational. In the tests we will take constant step sizes rather than variable ones. Variable step sizes are redundant here. The step size in the diffusion computation will be denoted by Δt_{dif} and taken equal to $\Delta t_{\text{split}}/4$. The extra factor of 0.5 compared to Δt_{adv} is needed because the integration formula is a two-step one.

3.6 Mass balance correction

The advection scheme, the implicit BDF integration formula and the spatial discretization of the vertical diffusion all guarantee conservation of mass. However, Gauss-Seidel iteration is not a mass conservative iteration process and since we approximate the implicit BDF solutions with only two iterations, the mass balance can become perturbed. A simple remedy to enforce conservation is to evaluate mass law expressions after each Gauss-Seidel iteration and to correct all species concentrations involved with the ratio (total mass before)/(total mass after). This correction is very cheap with regard to CPU time and can enhance the accuracy notably when the integration is carried out with large step sizes. In the type I splitting procedure the correction is carried out for each of the grid cells and in the type II procedure for each of the vertical columns. The chemistry model introduced in Section 4.1 conserves nitrogen. We therefore have applied the above mentioned mass balance correction for all nitrogen compounds.

3.7 Reducing the splitting error

At this stage of development we have discussed the full discretization for two operator splitting procedures. Operator splitting is attractive, but it induces a splitting error on top of all the discretization errors. By using Strang splitting, one achieves a reduction of this error, compared to that of the most simple splitting procedure. For pure advection-reaction equations the splitting error can even be eliminated [11, 7]. Spee [15] has implemented this elimination for a 2D advection-reaction problem which is related to the benchmark problem presented in Section 4. Thereupon we have used the elimination idea in the tests with the type II splitting procedure reported in the preprint [16]. However, in 3D it is theoretically justified only if the wind field is purely horizontal and constant in the vertical direction. Needless to say that this imposes a restriction in practice. We therefore have not implemented the elimination in the tests presented here. A second reason to abandon it is that it would interfere with the comparison between the two splitting procedures, as the elimination is not applicable to the type I splitting procedure.

4. A BENCHMARK PROBLEM FOR 3D GLOBAL TRANSPORT AND CHEMISTRY

To test the numerical schemes, to compare the two splitting procedures, and to measure vectorization/parallelization performances, we have developed a benchmark problem for 3D global transport and chemistry. This benchmark problem is similar to the regional test problem from [22] and when we omit the vertical diffusion, it is similar to the 2D problem from [15]. The complete description and a reference solution are available through World Wide Web ¹. To avoid confusion we note that in a number of respects the problem is different from the one presented in the preprint [16].

The problem contains horizontal advection, vertical diffusion and chemical reactions. Hence there is no vertical advection and no horizontal diffusion. Orography is not present either, nor do we use real meteorological data so that we are not confronted with massive I/O operations. We realize that this imposes a restriction. However, this restriction enables us to compute a very accurate reference solution which can be used for assessing numerical accuracy and efficiency.

4.1 The chemistry model

The chemistry model consists of 45 reactions between 17 species and is used in actual long term global studies where it is referred to as methane chemistry. We obtained it from [19]. This model is fully described in the appendix of the preprint to [23]. The reaction set gives rise to stiffness. The eigenvalues of the Jacobian matrix lie approximately between -10^9 and 0 (sec^{-1}). There are two extremely large eigenvalues which originate from the free radicals O^1D and O^3P . The chemistry is photochemical. Hence part of the reaction coefficients depend on the solar zenith angle, which depends on the time of the day and the location on earth. Part of the reaction coefficients also depend on the temperature and the pressure. This dependence is chosen in close accordance with the US Standard Atmosphere (1976). Emissions are not yet prescribed but otherwise the chemistry model is representative for what is currently used in global modeling, although there is a trend towards more complicated chemistry involving more species. All species are advected in the wind field and all species undergo vertical turbulent transport.

4.2 Initial condition and time interval

The unit of concentration is number of molec/ cm^3 and the unit of time is sec. Integration takes place over exactly 14 days, starting at midnight Greenwich Mean Time. With the exception of HNO_3 and NO , for all species the initial concentrations are at a uniform background level over the globe. Consequently, they are far from the diurnal photochemical equilibrium so that at the start of the integration strong initial transients will be encountered. The ground level values are given in Table 1 and initial concentrations in the other vertical levels are such that in each vertical column the mixing ratio (concentration/density) is constant. The initial concentrations of HNO_3 and NO are cylinder

¹http://www.cwi.nl/ftp/edwins/Ref_Sol_Benchmark_Global.html

shaped, see Figure 2. Needless to say that a cylinder shaped initial condition is a challenge for any advection scheme. At the initial time, the center of the cylinder lies at $(\lambda', \phi' = 0^\circ, 0^\circ)$. For HNO_3 and NO , the background concentration is $2.55 * 10^9$ and 100, respectively, and the maximum concentration is $4 * 10^9$ and 10^9 . Also for these two species the mixing ratio is taken constant over the vertical layers.

	Name	Species	Concentration
	-	M	$2.55 * 10^{19}$
	Water	H_2O	$2.55 * 10^{17}$
	Carbon monoxide	CO	$2.55 * 10^{12}$
	Molecular oxygen	O_2	$5.3295 * 10^{18}$
1	Nitric acid	HNO_3	see main text
2	Peroxynitric acid	HO_2NO_2	$1.0 * 10^2$
3	Nitrous acid	HNO_2	$1.0 * 10^2$
4	Hydroperoxide	H_2O_2	$1.0 * 10^2$
5	Ozone	O_3	$7.65 * 10^{11}$
6	Formaldehyde	$HCHO$	$1.0 * 10^2$
7	Methylhydroperoxide	CH_3OOH	$1.0 * 10^2$
8	Methylperoxy radical	CH_3O_2	$1.0 * 10^2$
9	Methane	CH_4	$4.335 * 10^{13}$
10	Nitrogen oxide	NO	see main text
11	Nitrogen dioxide	NO_2	$5.1 * 10^9$
12	Nitrate radical	NO_3	$1.0 * 10^2$
13	Hydroxyl radical	OH	$1.0 * 10^2$
14	Hydroperoxyde radical	HO_2	$1.0 * 10^2$
15	Dinitrogen pentoxide	N_2O_5	$1.0 * 10^2$
16	Atomic oxygen	O^1D	0.0
17	Atomic oxygen (g.s.)	O^3P	0.0

Table 1: Initial concentrations at ground level in [molec/cm³].

4.3 Advection

A divergence free, horizontal wind field in analytical form is used to enable the computation of an exact reference solution along characteristics, similar as in [15, 22]. Hence vertical advection does not take place. The wind field stems from [14] and describes a solid body rotation which takes exactly 14 days, see Figures 2 and 3. The velocities are given by

$$u = 2\pi\kappa(\cos\beta\cos\phi + \sin\beta\sin\phi\cos\lambda), \quad v = -2\pi\kappa\sin\beta\sin\lambda, \quad (4.30)$$

where $\beta = 45^\circ$ and $\kappa = a/(14 \times 24 \times 3600)$. Observe that the velocities are constant in the vertical direction and in time. The maximal velocity is approximately 125 km/hour. Near the surface this wind is of course very strong, but higher up in the atmosphere these velocities do occur.

4.4 Vertical diffusion

In real models the vertical turbulent diffusion coefficient K depends on the mixing height which is space and time dependent. We have used a constructed expression which simulates this dependence, such that at night $K \approx 0$, at noon a maximal value is reached while always $K \approx 0$ outside the planetary boundary layer. The function values for K at the equator location $(\lambda', \phi') = (0, 0)$ are shown in Figure 4. We see that a maximal value of $60 \text{ m}^2\text{s}^{-1}$ occurs at a height of about 700 m and

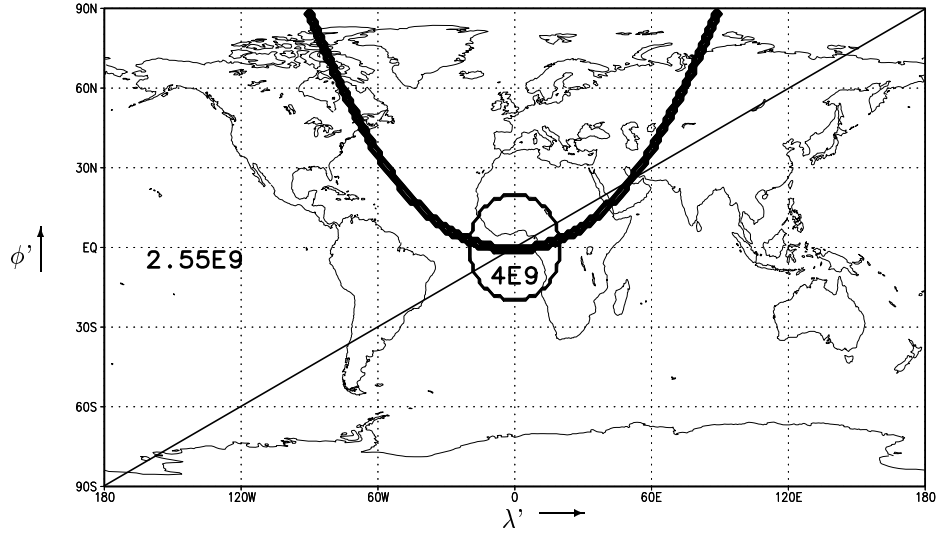


Figure 2: The initial concentration vector for HNO_3 and the trajectory of the cylinder. The coordinates are in degrees: $\lambda' = (\lambda - \pi) \frac{180}{\pi}$ and $\phi' = \phi \frac{180}{\pi}$.

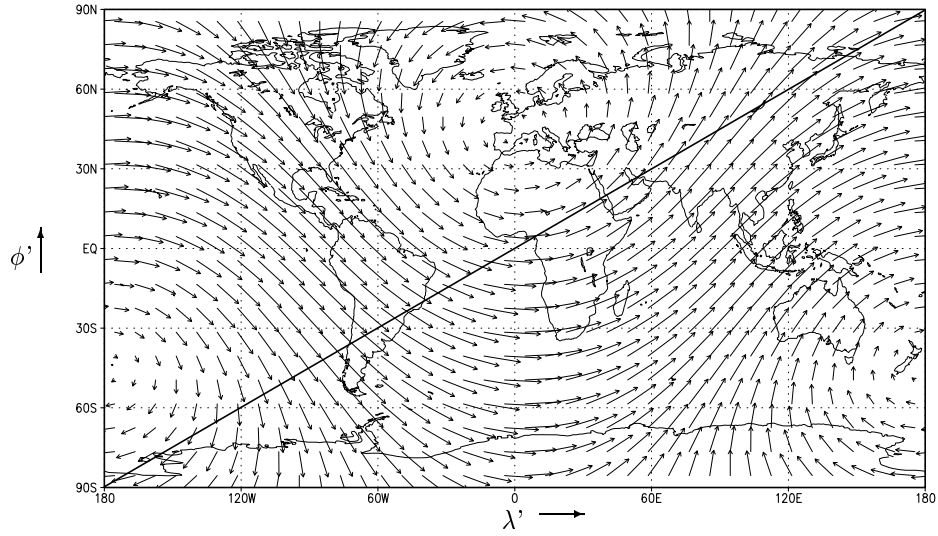


Figure 3: The prescribed wind-field ($\lambda' = (\lambda - \pi) \frac{180}{\pi}$; $\phi' = \phi \frac{180}{\pi}$).

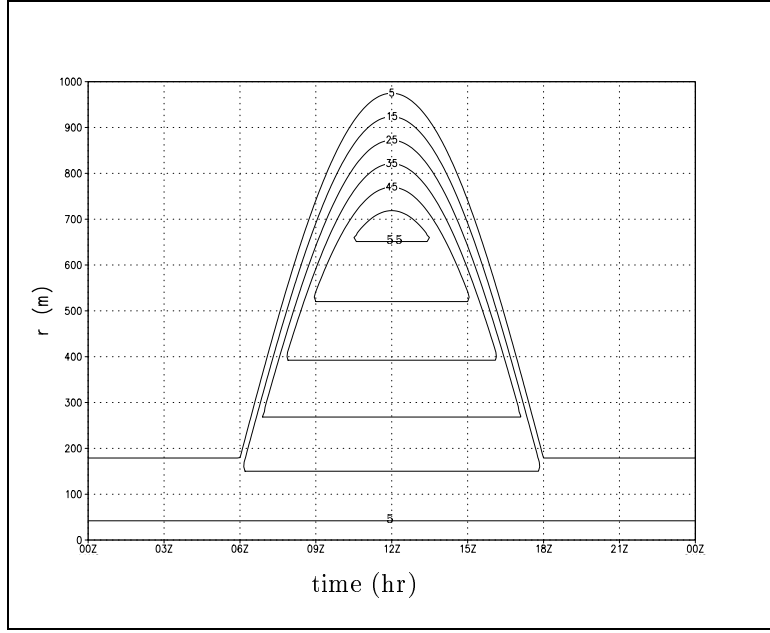


Figure 4: K in $m^2 s^{-1}$ as a function of the time and vertical coordinate r .

that upward K diminishes to $5 m^2 s^{-1}$ at about 1000 m height. Higher up K further diminishes to zero.

We have used 15 cell-centered layers to approximate the vertical diffusion term. Simulating a pressure-based vertical grid, the distribution of the cell-centers is a function of the pressure which is taken uniform over the globe. The lowest cell-center, cf. (3.20), lies at sea level (1000 hPa) and the highest at 22.6 km (0 hPa). The complete distribution reads 0.0, 0.7, 1.3, 2.6, 4.9, 6.3, 8.0, 9.0, 10.3, 11.8, 13.8, 15.4, 16.9, 18.9, 22.6 km. The grid is rather coarse in the boundary layer and therefore only 3 cell-centers are positioned there.

A characteristic value to assess the numerical stability of a time stepping method for our diffusion problem, is the dimensionless product of the step size Δt and the spectral radius, denoted by σ , of the matrix \mathbf{A} introduced in equation (3.24). An upperbound for σ is found by applying Gerschgorin's theorem, which in the present situation yields

$$\Delta t \sigma \approx \Delta t \max \frac{4K}{(\Delta r)^2}, \quad (4.31)$$

where the coefficient K and the grid distance Δr vary over the grid. If $\Delta t \sigma \approx 1$, then explicit time stepping will do for numerical stability. If this product is $\gg 1$, however, an implicit approach is necessary. Our current choice for the grid and the coefficient K yields

$$\Delta t \sigma \approx \Delta t \frac{4 \times 60}{600^2} = \frac{\Delta t}{1500}, \quad (4.32)$$

which is fairly small in the sense that even a step size of approximately 0.5 hour still allows an explicit treatment. Obviously, if in the boundary layer a much finer grid would be used, like in regional and urban models, then the stability restriction can be much more severe.

In both splitting procedures we treat the diffusion term implicitly. Since in our approach this gives rise to the solution of tridiagonal linear systems only, the additional costs compared to an explicit

treatment are not high. For other methods solving chemistry and vertical diffusion in a coupled manner, the above considerations can have a greater impact for the efficiency of treating vertical diffusion explicitly or implicitly. Needless to say that the range of values K takes on should also be taken into consideration. Our maximum value of $60 \text{ m}^2\text{s}^{-1}$ seems rather high, compared to for example the maximum value of 10 used in [5]. In other words, it may well be that in actual global models, where the vertical grid sizes in the planetary boundary layer will not be smaller than 100 m, say, and the maximum for K is not greater than $10 \text{ m}^2\text{s}^{-1}$, an explicit treatment can be used without a severe time step restriction.

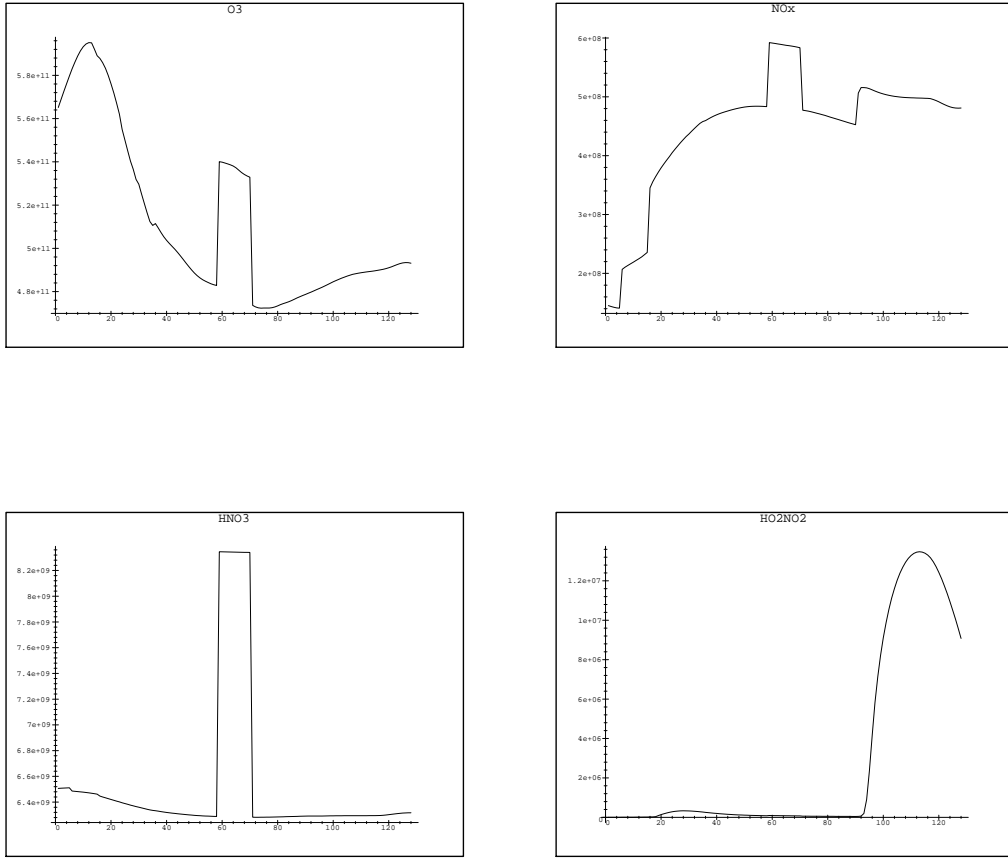


Figure 5: Reference solutions at ground level of O_3 , NO_x , HNO_3 and HO_2NO_2 in unit molec/cm^3 (vertical axis) plotted along the diagonal $\lambda' = \phi'/2$ (horizontal axis).

4.5 The reference solution

In case of pure advection, the cylindrically shaped initial profiles for HNO_3 and NO would be transported over the northern hemisphere as depicted in Figure 2 and return at their initial position at the diagonal line $\lambda' = \phi'/2$ in exactly 14 days. We therefore have computed an accurate reference solution at all vertical layers lying above this whole line. Obviously, vertical diffusion and chemistry change

the solution profile. But still the test is quite meaningful for advection schemes.

Without vertical diffusion, computing an accurate reference solution can be done by backward solving the chemistry along characteristics. To circumvent the difficulty of vertical diffusion, which prevents this, we have followed this procedure for the semi-discrete system which results from discretizing the diffusion operator, similar as in [22]. Hence our reference solution can be considered exact except for the error due to the spatial discretization of the vertical diffusion term. This means that assessment of accuracy is possible only for the horizontal advective transport, the solution of the chemistry and the time integration of the vertical diffusive transport.

Figure 5 shows the reference solution for O_3 , NO_x , HNO_3 and HO_2NO_2 along the diagonal line at ground level. One can see that a rather fine grid is needed to resolve all details and that as a result of the chemical reactions a cylindrical profile is also present for species other than HNO_3 and NO .

4.6 Problem size

To illustrate the scale of the numerical problem, we included Table 2 which for three practical choices of the spatial grid gives the total number of concentration values that must be computed by integration in time. In this table we have taken into account the savings of the grid reduction near the poles. The total number of unknowns clearly illustrates that global air quality modeling leads to a huge computational task. In this connection it is important to recall that our time span of 14 days is short and that our chemical scheme composed of 17 species is of moderate size, see [12]. The same can be said about the space grids. The finest longitude/latitude grid here is approximately $1.4^\circ \times 1.4^\circ$. In [12] it is pointed out that still finer grids are needed in future generation models.

Table 2: Problem size.

Grid (λ, ϕ, r)	# uniform grid cells	# unknowns on uniform grid	# grid reductions	# reduced grid cells	# unknowns on reduced grid
$64 \times 32 \times 15$	30.720	522.240	4	24.840	422.280
$128 \times 64 \times 15$	122.880	2.088.960	5	93.960	1.597.320
$256 \times 128 \times 15$	491.520	8.355.840	6	391.560	6.656.520

5. VECTORIZATION, PERFORMANCE, ACCURACY AND EFFICIENCY

Our tests have been carried out on a Cray C90. It thus is of interest to optimize the code with respect to vectorization. In this section we outline the vectorization, we present results of performance tests on the benchmark problem and comment on the achieved numerical accuracy and efficiency.

5.1 Vectorization

On a uniform grid, vectorization of the explicit advection computation is straightforward and can easily be optimized. In [1] it is shown that the non-uniformity near the poles only leads to a minor reduction of the vector speed, mainly because the larger part of the grid is still uniform. However, even on the non-uniform part of the grid the vector speed is still quite acceptable. For the technicalities and modifications required near the poles we refer to [1].

The vectorization of the coupled chemistry-diffusion computation in the type II splitting procedure is similar to that in [22] and is carried out along the horizontal grid dimensions. In a different setting, this idea of vectorization was first proposed in [4]. As far as we know, in air quality modeling it was first reported in [8]. Here it works because the chemistry-diffusion computation results in a coupling only in the vertical direction, which means that all operations involved can be executed simultaneously over the horizontal grid points (λ_i, ϕ_j) .

However, as pointed out in [8], there is a snag in the choice of the step size Δt_{evd} . For grid-vectorization we would like to use one and the same integration step size over the horizontal grid, because then the vector length would always be sufficiently large for reaching the optimal vector speed.

Unfortunately, this would cause a reduction of efficiency, because at parts of the globe we have to obey step size restrictions more severe than elsewhere, viz. in the (constantly changing) areas on the globe where day turns into night and vice versa. In these areas the species undergo intense chemical reactions since the chemistry is photochemical. This gives rise to rapid changes in concentrations. On the other hand, during nightly periods and also during day time, changes are much slower, so that adjusting the step size to the sunset/sunrise situation everywhere, is less efficient. As a compromise, we therefore group points (λ_i, ϕ_j) in clusters of length 128, the vector length of the C90. This grouping of points into clusters is in the ϕ -direction, in order to stay in the same time zone as much as possible. The chemistry-diffusion computation is then performed and vectorized for a single cluster using the same step size Δt_{cvd} for this cluster, while step sizes may differ per cluster. This compromise works out satisfactorily, although we will still encounter step sizes that are too small for part of the grid points. We emphasize that our clustering technique differs from the one used in [8]. In [22] clustering has not been considered.

The vectorization of the diffusion computation in the type I splitting procedure goes entirely similar, i.e., also along the horizontal grid dimension. But the clustering issue is of no relevance here, as constant step sizes Δt_{dif} are used. The chemistry computation does use clustering though, since the step size Δt_{che} is variable. Because only box models appear, the clustering can now be carried out also in the vertical direction so that the clustering region in the horizontal plane can be kept much smaller.

It is also possible to do part of the computational work simultaneously over the vertical direction as well, e.g. the computation of the production terms, which offers a way of increasing the vector length. In order to realize this profit we implemented separate subroutines for performing essentially the same computations for the uniform and the non-uniform part of the grid. So-called (Cray) CDIR-directives were not needed because we could benefit from the loop collapsing (i.e. merging nested loops into one single loop) done by the (Cray) FORTRAN preprocessor (FPP).

5.2 Performance results

We ran our code, written in FORTRAN 77, on the Cray C90 at SARA, Amsterdam. Timing results presented in this section were done on one processor with a clock cycle time of 4.2 nanosecond and a double vector pipe. This gives a theoretical peak performance on one processor of 476 Mflop per second and 952 when chaining an add and a multiply. Chaining floating point operations, however, can only be achieved for very specific loops so that in practice a performance of 50% of the peak performance (500 Mflops) can already be considered as a very good result. To measure the Mflop rate and the CPU time of a routine, we used the Cray utility **perftrace** that gives the hardware performance by program unit.

The discretization parameters were chosen as follows. Three different reduced longitude-latitude grids were used with the same vertical grid, see Table 2. These grids determine the critical advection step size Δt_{adv} for stability and positivity through the CFL condition (3.17). The critical values are, approximately, 40, 20 and 10 min. In our tests these values determine $\Delta t_{\text{split}} = 2 \Delta t_{\text{adv}}$ and $\Delta t_{\text{dif}} = \Delta t_{\text{adv}}/2$. The step sizes Δt_{che} in splitting I and Δt_{cvd} in splitting II vary in time and are governed by the step size control mechanism of the integration method. Both are constrained by a minimum of 5 min. throughout. This minimum step size was selected after a few trial and error runs. This minimum value is important as it determines to a great extent the CPU time and the accuracy of the chemistry integration.

Table 3 shows performance results for the two splitting procedures on the three different grids. The values for Δt_{cvd} and Δt_{che} given in this table are average values. For the two procedures as a whole we do not observe large performance differences. Therefore we will only comment on the type II procedure, since this procedure performs significantly better with respect to accuracy (see Section 5.3).

The overall Mflop rate is close to 500 which means close to 50% of the peak formance. On the coarsest grid the Mflop rate of 370 in the advection computation is low due to the grid reduction. The finer the grid, the smaller the influence of grid reduction will be so that larger flop rates are obtained.

When comparing CPU times needed for the advection and the chemistry-diffusion computation, we see that on the coarsest grid the latter is three times more expensive, while on the finest grid the advection takes 44% of the total CPU time versus 56% for the chemistry-diffusion computation. The advection becomes more costly due to the diminishing step size Δt_{adv} , whereas the step size Δt_{cud} of the chemistry-diffusion computation hardly changes and stays close to the imposed minimum of 300 sec. Three reasons for this behaviour can be mentioned. First, we always start with the minimum of 300 sec. which determines a certain maximum value. For example, on the finest grid, where the split interval is only 1200 sec. long, the smallest possible number of steps with our two-step method equals three so that the largest possible average for Δt_{cud} is only 400 sec. ($= (300 + 300 + 600)/3$ sec.). Of course, on the coarsest grid there is more time for Δt_{cud} to increase. Second, the initial transients introduced at the beginning of each split interval do withhold the step size from increasing very rapidly directly from start on. As the third reason we recall the global nature of the problem which means that we always have to integrate through sunsets and sunrises during which rapid temporal changes in concentration values exist.

The imposed tolerance for the step size control also plays a role in the above. We have not examined this issue further, since a step size of ≈ 5 min. for a low cost chemistry-diffusion computation is already quite efficient, in our opinion. It is obvious, though, that we might as well have used a constant step size of 5 min. throughout which makes the cluster approach redundant and also avoids the costs of the step size control.

5.3 Accuracy

For the six tests tabulated in Table 3, Figures 6 - 9 show plots of the computed and reference solution concentrations of O_3 , NO_x , HNO_3 and HO_2NO_2 in the same way the reference solutions were depicted in Figure 5.

For the chosen discretization parameters, the type II splitting (cases (b1) - (b3)) can be seen to work very well. Even the coarsest grid results (cases (b1)) are already reasonably accurate for the actual practice, while going to a finer grid (cases (b2) and (b3)) clearly resolves all details in the solutions. Noteworthy is the excellent performance of the advection scheme in resolving the cylindrical profiles.

The type I splitting (cases (a1) - (a3)) falls behind, as we anticipated before (see also [5] for a similar conclusion). Even on the finest grid the errors are significant and in fact larger than on the coarsest grid. When comparing the finest and coarsest grid resolution, one sees that the spatial error gets smaller upon spatial refinement. This means that the source of the errors, and their growth, lies in the time stepping process. Indeed, to a great extent the errors emanate from the chemistry integration in which a minimal step size of 5 min. is used on all grids. While 5 min. is small enough in the coupled procedure, it is much too large for the uncoupled one. Apparently, splitting chemistry and vertical diffusion strongly disturbs chemical equilibria resulting in strong initial transients. A step size of 5 min. is then too large to resolve these sufficiently accurately over the split intervals used. Since on the coarsest grid the split interval is 4 times longer, the errors are also smaller simply because the initial error has been decreased stronger. Would we use a 1200 sec. split interval also on the coarsest grid, the errors would become close to those of the finest grid shown in the figures. A separate test has confirmed this.

To further illustrate that to a great extent the errors emanate from the chemistry integration, we have repeated the test (a1) with (at least) a tenfold smaller, constant chemistry step size Δt_{che} of 0.5 min., while the other step sizes were left unchanged. Figure 10 shows the results for O_3 and HO_2NO_2 . One sees that indeed the errors have become much smaller, but the accuracy is not yet as good as in case (b1) for the type II method. In a second test we then have reduced the other step sizes with a factor of 4, using again $\Delta t_{che} = 0.5$ min, to further reduce all other temporal errors present in the computation. Even then the accuracy for (b1) is still better, see again Figure 10. This clearly shows the superiority of the type II method.

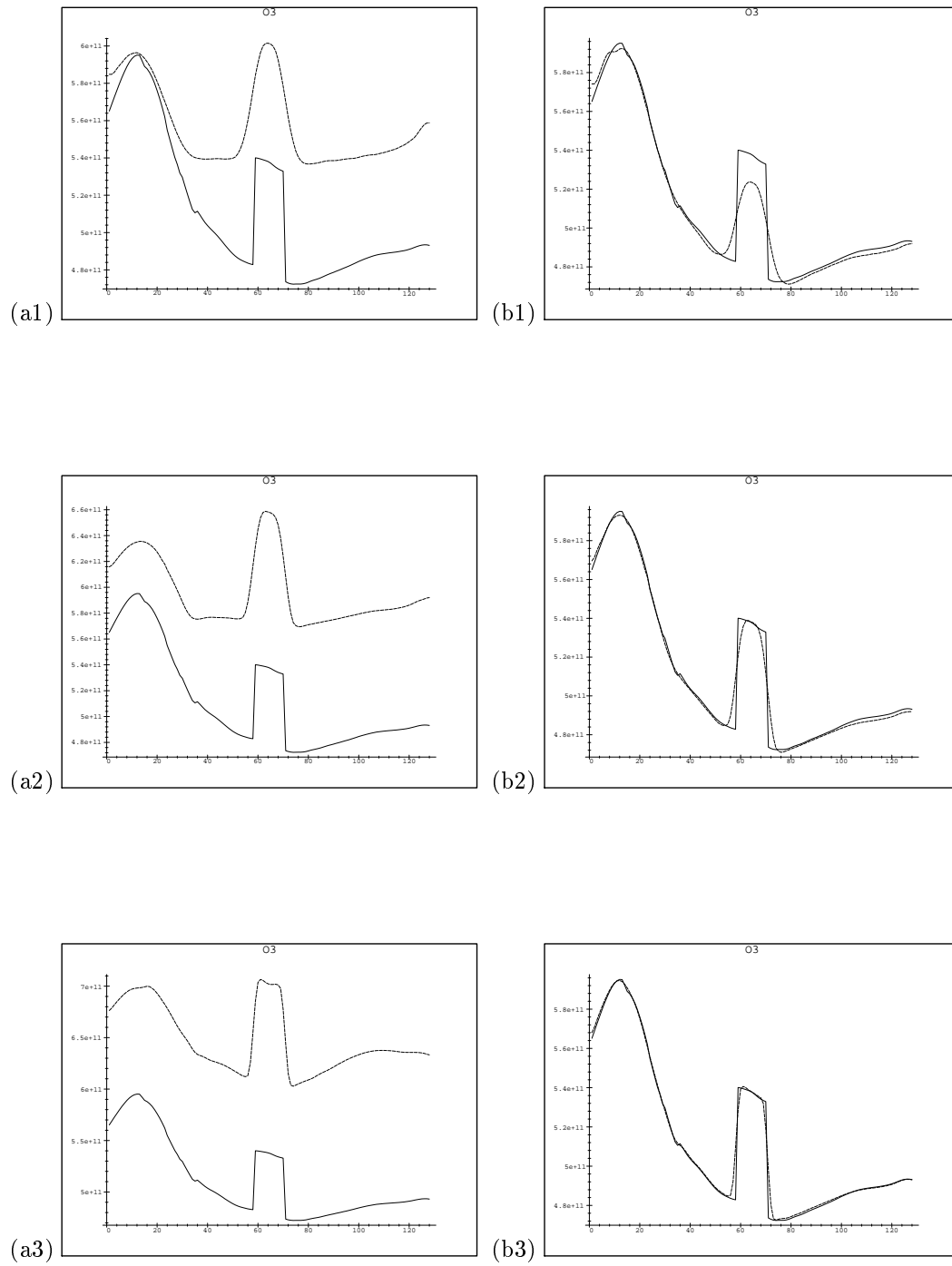


Figure 6: Reference solution and computed solution of O_3 . The left column shows results for the type I splitting and the right column for the type II splitting. The two upper plots are for the coarsest grid, etc. The vertical scaling differs per plot.

Table 3: Performance on one processor of the C90.

(a) Results for the type I splitting procedure.

Grid (λ, ϕ, r)	Advection			Diffusion			Chemistry		
	Δt_{adv} (secs.)	CPU %	rate (Mflops)	Δt_{dif} (secs.)	CPU %	rate (Mflops)	Δt_{che} (secs.)	CPU %	rate (Mflops)
$64 \times 32 \times 15$	2400	29	364	1200	9	259	306	62	413
$128 \times 64 \times 15$	1200	33	483	600	13	259	300	53	410
$256 \times 128 \times 15$	600	40	522	300	19	240	300	41	383

Grid (λ, ϕ, r)	Total	
	rate (Mflops)	CPU (secs.)
$64 \times 32 \times 15$	383	500
$128 \times 64 \times 15$	413	2410
$256 \times 128 \times 15$	412	15500

(b) Results for the type II splitting procedure.

Grid (λ, ϕ, r)	Advection			Chemistry-diffusion		
	Δt_{adv} (secs.)	CPU %	rate (Mflops)	Δt_{cvd} (secs.)	CPU %	rate (Mflops)
$64 \times 32 \times 15$	2400	26	370	306	74	488
$128 \times 64 \times 15$	1200	34	484	340	65	464
$256 \times 128 \times 15$	600	44	523	318	56	425

Grid (λ, ϕ, r)	Total	
	rate (Mflops)	CPU (secs.)
$64 \times 32 \times 15$	455	545
$128 \times 64 \times 15$	470	2340
$256 \times 128 \times 15$	468	14100

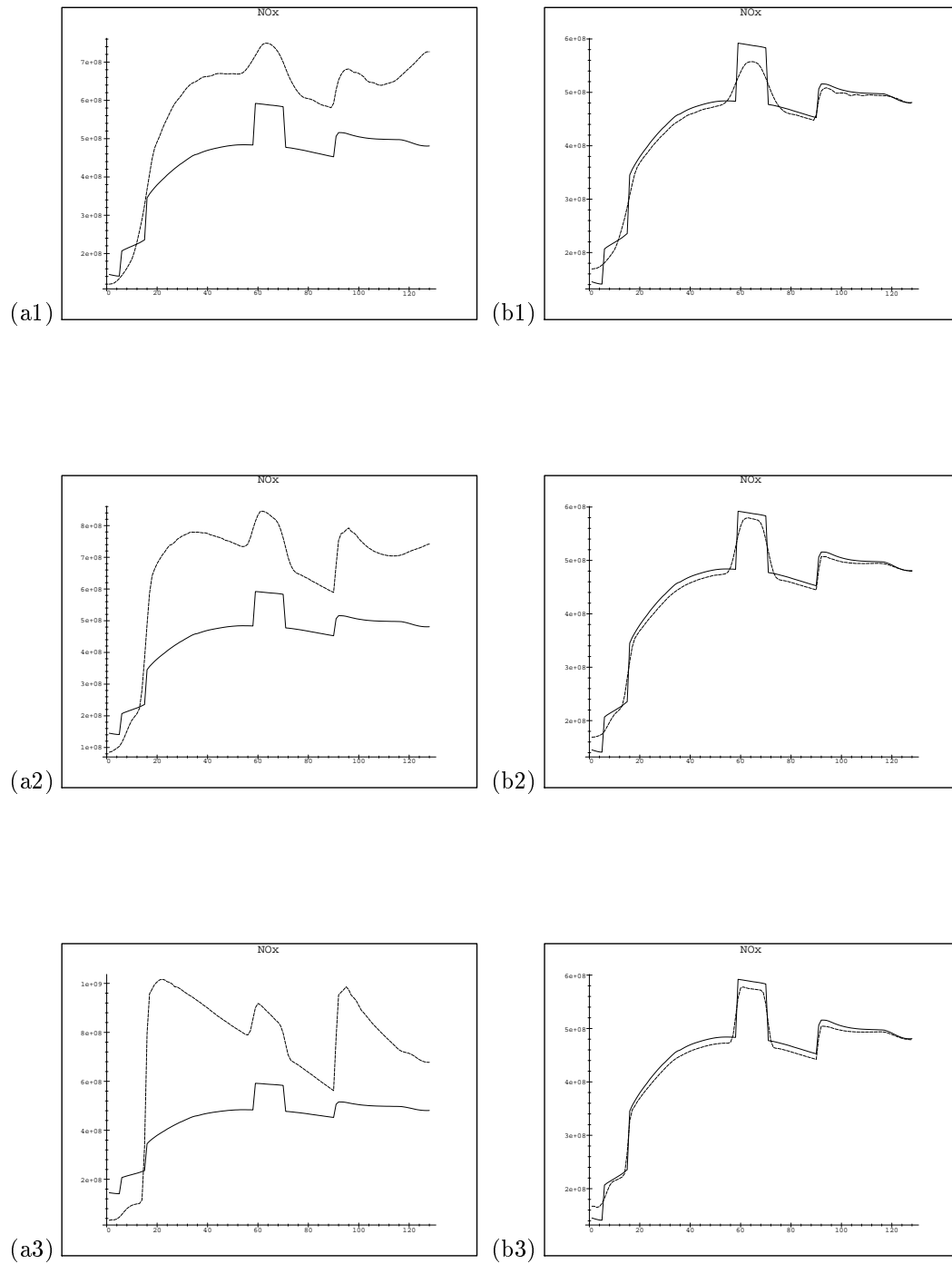


Figure 7: Reference solution and computed solution of NO_x . The left column shows results for the type I splitting and the right column for the type II splitting. The two upper plots are for the coarsest grid, etc. The vertical scaling differs per plot.

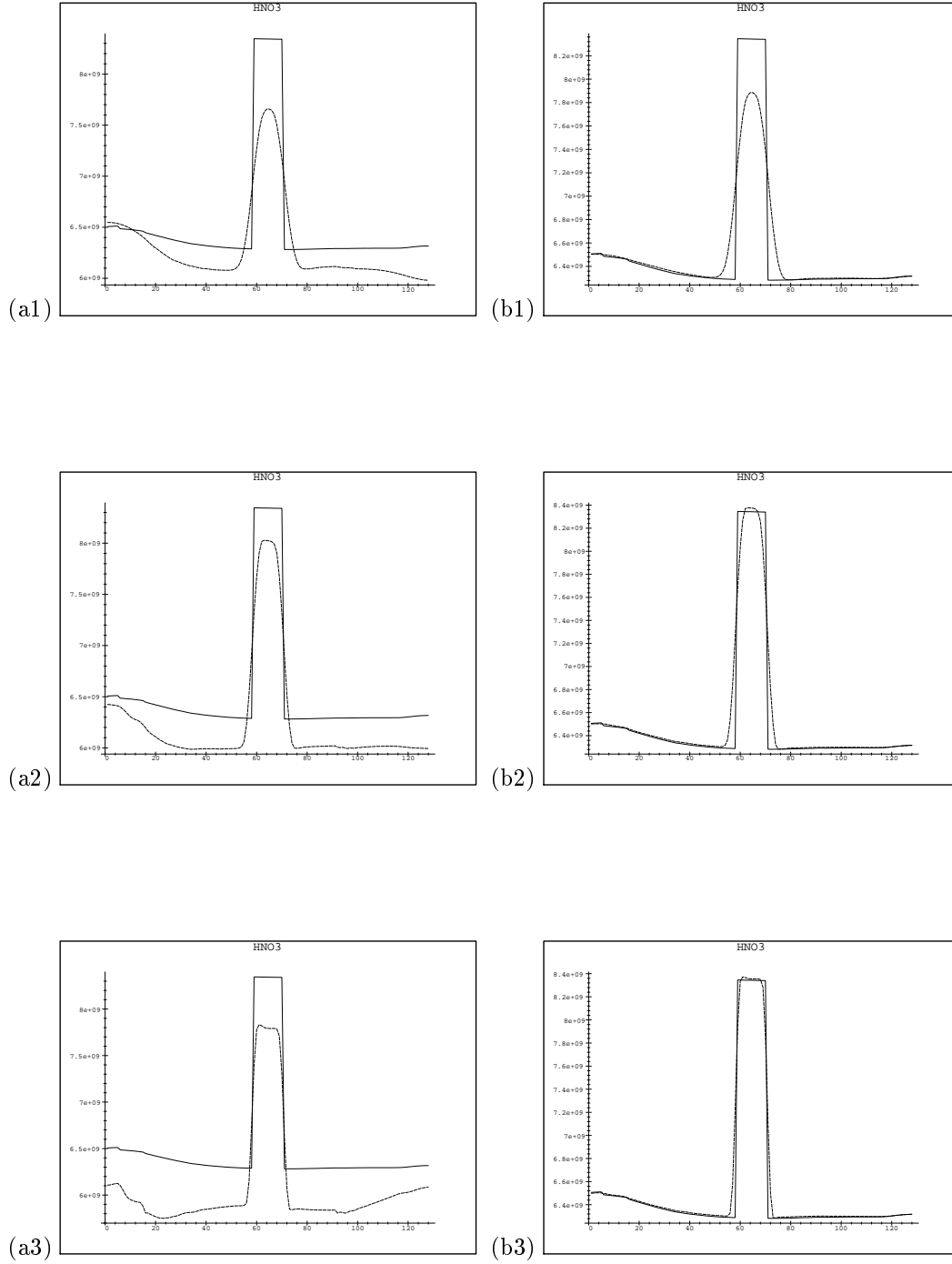


Figure 8: Reference solution and computed solution of HNO_3 . The left column shows results for the type I splitting and the right column for the type II splitting. The two upper plots are for the coarsest grid, etc. The vertical scaling differs per plot.

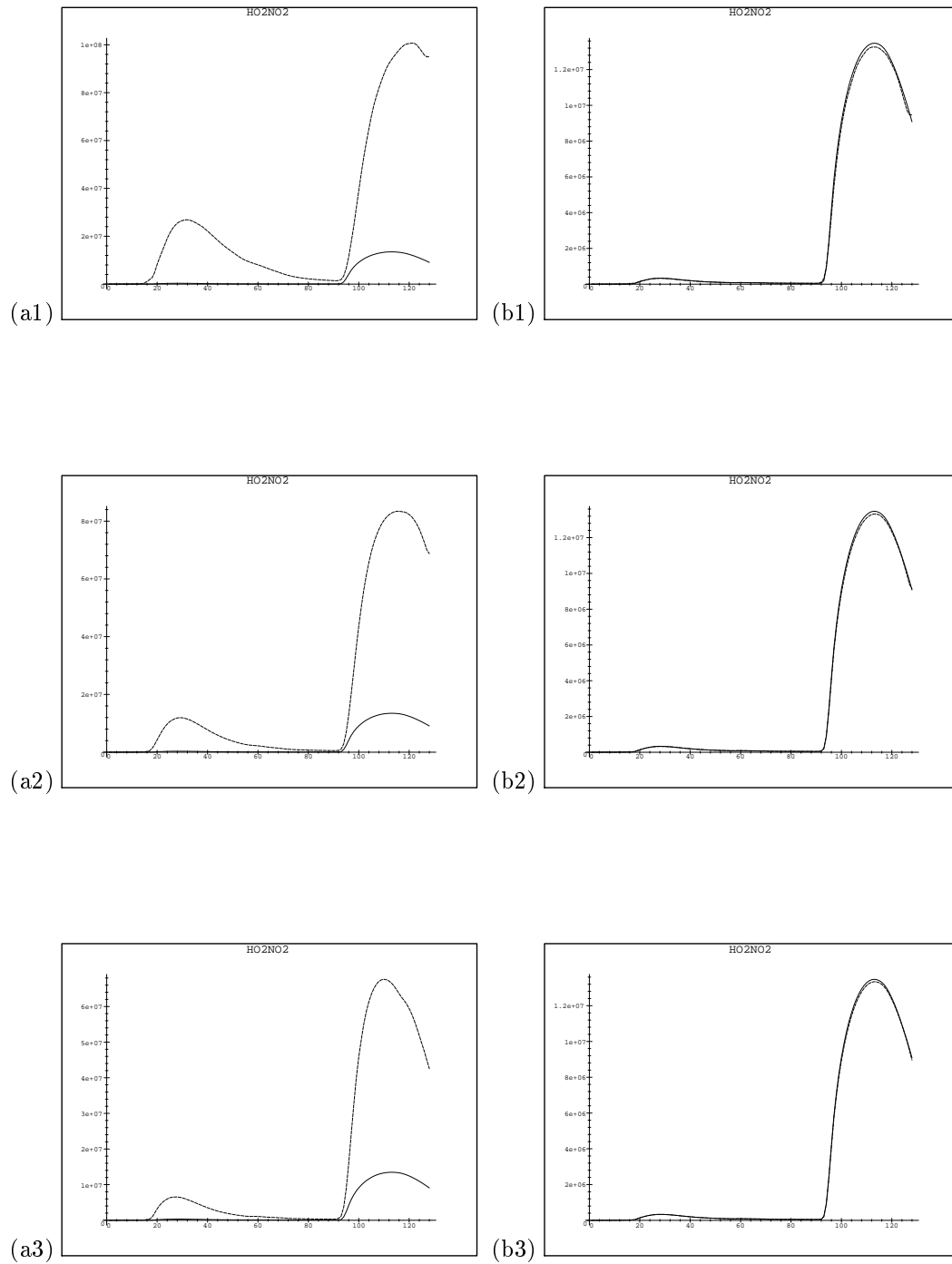


Figure 9: Reference solution and computed solution of HO_2NO_2 . The left column shows results for the type I splitting and the right column for the type II splitting. The two upper plots are for the coarsest grid, etc. The vertical scaling differs per plot.

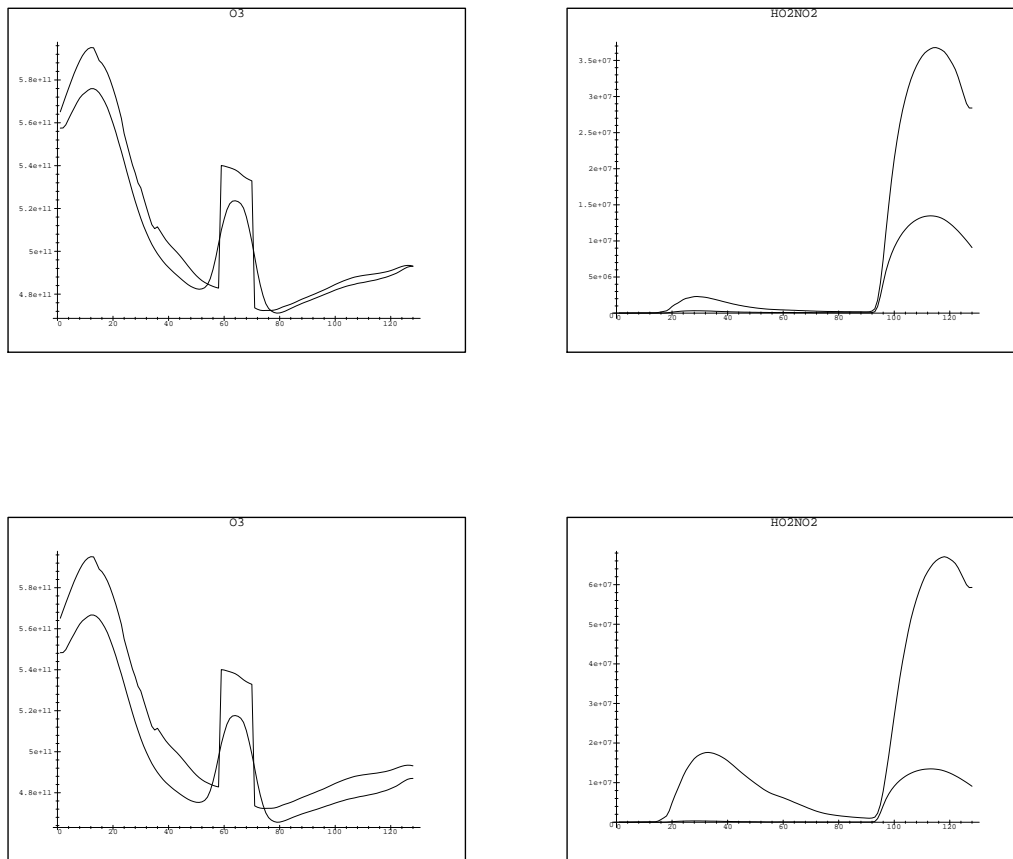


Figure 10: Reference solution and computed solution of O_3 and HO_2NO_2 , illustrating the splitting error in case (a1). The two plots above illustrate the first test and the two lower ones the second.

5.4 Efficiency

Efficiency and the required CPU times are more difficult to assess since we cannot compare our results with those obtained by other solvers. However, the following observation indicates that our splitting method II and its vectorized implementation are quite efficient. Table 3 shows that the CPU time for the advection computation on the finest grid is only 12% less than for chemistry and vertical diffusion together. This means that chemistry and diffusion are dealt with efficiently when taking the advection computation as a reference point. Of course, the accuracy should be sufficiently high to render this conclusion of practical value. We have shown that this is indeed true for the current benchmark problem.

It is also of some interest to compare simulation time/CPU time ratios for our benchmark tests with the predictions given in Table 3 of the review paper [12]. For a hypothetical model, representative for the state of the art, that table predicts required flop rates for three different grids and three different ratio's. For our $1.4^\circ \times 1.4^\circ \times 15$ grid computation, lower table (b), we have a ratio of 85:1 against ≈ 0.5 Gflops. Comparing this ratio with the ones given in [12], reveals a wide gap, clearly to the

advantage of our solver. For example, to achieve a 100:1 ratio for a grid resolution of $2^o \times 2^o \times 10$, the flop rate prediction in [12] is 20 to 30 Gflops, whereas we are close to this ratio with ≈ 0.5 Gflops on a finer grid. This again indicates that our solver is efficient. However, care is needed here as there are many uncertainties in this comparison, an important one being the computational complexity of our benchmark compared to that of the hypothetical problem of [12].

6. PARALLELIZATION ON THE C90

While vectorization effectively reduces CPU time and takes place on a single processor, parallelization merely reduces the wall-clock time by distributing the work over multiple processors. In this section we discuss the parallelization of the type II splitting procedure, for which two different approaches have been considered, viz. autotasking and a parallelization over the geometry.

Autotasking can be described as the automatic distribution of loop iterations to multiple processors [3]. This type of parallelization takes place on the level of elementary algebraic operations and dependency analysis across procedure boundaries does not take place. Autotasking takes a FORTRAN program as input, then modifies it and adds compiler directives, so that it can run concurrently on multiple processors. This works best on programs in which most of the work concerns independent operations in nested do-loops. If possible, the innermost loop of a nest of do-loops is vectorized and autotasking runs the outermost loop on multiple processors. In some cases, autotasking will process a single vectorizable loop into chunks.

Because the advection computation is explicit, for this part we can rely on autotasking (the explicit Runge-Kutta method invokes merely elementary algebraic operations on long vectors). Autotasking of the chemistry-diffusion problem is less efficient, however. For this part we have implemented an alternative which, conceptually, is a simple form of domain decomposition. Recall that for the grid-vectorization we group points (λ_i, ϕ_j) on the globe in clusters of length 128. But now we have p processors available instead of only one. Hence we can distribute clusters over the processors and use, on each processor, the same vectorized chemistry-diffusion computation. This way of parallelization is much more effective than autotasking. So-called (Cray) CMIC-directives were used to assign each cluster to a different processor. By such directives we could parallelize loops containing calls to subroutines which are on a high algorithmic level. One needs to be careful though and indicate clearly which variables are global data and which are local.

Load balancing refers to the even distribution of work over all processors. Here, we need to reckon with a step size Δt_{evd} that varies from one cluster to another. Therefore, one cluster may take more time steps within a splitting step than another one. This may affect the load balancing. Nevertheless the average effect appears to be moderate.

Regarding our expectations on parallel speed up, we have to bear in mind the restriction put by Amdahl's law. Suppose we have a number of p processors available for parallel execution of a code. Let $S = T(1)/T(p)$ denote the speed up with $T(p)$ the wall-clock time required to execute the code on p processors. Amdahl's law then reads

$$S = \frac{1}{(1 - f) + f/p}, \quad (6.33)$$

where f denotes the fraction of the work that can be executed in parallel. The effects of parallelization can be measured by means of the Cray tool **atexpert**. This tool predicts speedups on an almost dedicated machine from data collected from a run on a non-dedicated machine [3]. Table 4 contains the predicted values for the three test cases (b1) - (b3) of Table 3, assuming 2, 4 and 8 processors. The bracketed numbers correspond to the ideal speedup (or parallel fraction f) according to Amdahl's law (6.33). The results are self-evident. For example, on the coarsest grid the actual speedup for 8 processors is predicted to be 4 and on the finest grid 6.6. These expected speedups are very satisfactory when taking into account that the grid dimensions have not been adjusted to the number of processors. The corresponding parallel fraction for these two cases amounts to 93% and 99%, respectively. Finally, the predictions are in accordance with those from [22] (Table 3).

Table 4: Speedup factors for type II splitting.

Grid(λ, ϕ, r)	p = 2	p = 4	p = 8
$64 \times 32 \times 15$	1.8 (1.9)	3.0 (3.3)	4.0 (5.3)
$128 \times 64 \times 15$	1.8 (2.0)	3.2 (3.8)	5.6 (7.0)
$256 \times 128 \times 15$	1.9 (2.0)	3.6 (3.9)	6.6 (7.6)

7. CONCLUSIONS, FINAL REMARKS AND FUTURE WORK

Air quality modeling on a global scale is numerically extremely expensive and requires fast algorithms and sophisticated numerical software on high-speed computers. Using a powerful, shared memory, vector/parallel computer, a Cray C90, we have compared and tested two operator splitting procedures for a problem that we would like to propose as a benchmark for 3D global transport-chemistry solvers. There is a great need for model problems in this field. Without representative large-scale model problems, comprehensive testing is hardly possible, let alone comparison and validation of solvers. The interested reader is invited to join us in this benchmark activity and to apply his own solver to our test problem ². The present test results justify the following conclusions:

- The standard type I splitting procedure, where all processes are solved sequentially, in particular the stiff chemistry and the vertical turbulent diffusion, severely complicates the chemistry integration. Splitting stiff chemistry and vertical turbulent diffusion, processes which both possess small and large time constants, severely perturbs chemical equilibria resulting into strong initial transients within each split time step. These transients render the chemistry integration expensive, irrespective the integrator used.
- We advocate to keep the stiff chemistry and vertical turbulent diffusion coupled, as this alleviates the onset of transients. Yet, in spite of the 1D nature, a coupled solution can be rather expensive when standard modified Newton iteration is used within an implicit integration approach. Our type II splitting procedure is special in that it solves the stiff chemistry coupled with the vertical diffusion using a tridiagonal, Gauss-Seidel iteration. This iteration technique reduces the costs of the coupled approach significantly and results, together with the chosen advection scheme, in an efficient procedure for tropospheric gas-phase transport-chemistry models.
- Significant effort has been put in vectorizing the algorithms on the C90. We have shown that vectorization along the horizontal grid dimension is close to optimal. On fine grids the overall Mflop rate is about 500. This means that also the explicit horizontal advection scheme, which makes use of a reduced grid near the poles to alleviate the stability restriction, vectorizes very well.
- These high flop rates were measured on one processor. We also examined parallelization of the type II splitting procedure and have shown that by a rather straightforward approach, a very satisfactory speedup is realized. On fine grids, as considered for future practical applications, 75% of the optimal speedup seems within reach on the shared memory, vector/parallel C90 machine.

In the near future our research will be continued into two directions. First, a second benchmark problem will be considered involving real meteorological data and orography. The main purpose hereby is to further test and compare our reduced-grid advection scheme under more realistic conditions. A report on this investigation is already in preparation [17]. Second, alternative splittings will be examined which reduce the initial transient complications in the chemistry integration. One of our aims hereby is to further benchmark the recently proposed sparse Rosenbrock solvers [13] for application

²http://www.cwi.nl/ftp/edwins/Ref_Sol_Benchmark_Global.html

to transport-chemistry problems.

REFERENCES

1. J.G. BLOM, W. HUNSDORFER AND J.G. VERWER: Vectorization aspects of a spherical advection scheme on a reduced grid, Report NM-R9418, CWI, Amsterdam (1994).
2. G.R. CARMICHAEL, A. SANDU, F. POTRA, V. DAMIAN-IORDACHE AND M. DAMIAN-IORDACHE: The current state and the future directions in air quality modeling, in: *Modelling and Simulation of Complex Environmental Problems*, A. Sydow, G.R. Carmichael and G. Korn, eds., Proceedings of a Dagstuhl Seminar, Springer-Verlag (1996).
3. CRAY RESEARCH, INC.: Parallel Processing Guide, CF77 Compiling System, Volume 4, Cray Research, Inc. (1991).
4. G.H. GOLUB AND C.F. VAN LOAN: *Matrix Computations*, (John Hopkins University Press, Baltimore; second edition, 1990).
5. J. GRAF AND N. MOUSSIOPOULOS: Intercomparison of two models for the dispersion of chemically reacting pollutants, *Beitr. Phys. Atmosph.* 64 (1991) 13–25.
6. W. HUNSDORFER, B. KOREN, M. VAN LOON AND J.G. VERWER: A positive finite-difference advection scheme, *J. Comput. Phys.* 117 (1995) 35–46.
7. W. HUNSDORFER AND J.G. VERWER: A note on splitting errors for advection-reaction equations, *Appl. Numer. Math.* 18 (1995) 191–199.
8. M.Z. JACOBSON AND R.P. TURCO: SMVGEAR: A sparse-matrix, vectorized gear code for atmospheric models, *Atmospheric Environment* 28 (1994) 273–284.
9. O. KNOTH AND R. WOLKE: *A comparison of fast chemical kinetic solvers in a simple vertical diffusion model*, Proceedings 20th International Meeting on Air Pollution Modelling and its Applications, Valencia, Spain (1993).
10. B. KOREN: A robust upwind discretization method for advection, diffusion and source terms, in: *Numerical Methods for Advection-Diffusion Problems*, C.B. Vreugdenhil and B. Koren, eds., Notes on Numerical Fluid Mechanics 45 (Vieweg Verlag, Braunschweig, 1993) 117–138.
11. R.J. LEVEQUE AND H.C. YEE: A study of numerical methods for hyperbolic conservation laws with source terms, *J. Comput. Phys.* 86 (1990) 187–210.
12. L.K. PETERS, C.M. BERKOWITZ, G.R. CARMICHAEL, R.C. EASTER, G. FAIRWEATHER, S.J. GHAN, J.M. HALES, L.R. LEUNG, W.R. PENNELL, F.A. POTRA, R.D. SAYLOR AND T.T. TSANG: The current state and future direction of Eulerian models in simulating the tropospheric chemistry and transport of trace species: a review, *Atmospheric Environment* 29 (1995) 189–222.
13. A. SANDU, J.G. VERWER, J.G. BLOM, E.J. SPEE AND G.R. CARMICHAEL: Benchmarking stiff ODE solvers for atmospheric chemistry problems II: Rosenbrock solvers, CWI Report NM-R9614. This report has also been released in the report series *Reports on Computational Mathematics* of the University of Iowa (1996).
14. P.K. SMOLARKIEWICZ AND P.J. RASCH: Monotone advection on the sphere: an Eulerian versus semi-Lagrangian approach, *Journal of the Atmospheric Sciences* 48 (1991) 793–810.
15. E.J. SPEE: Coupling advection and chemical kinetics in a global atmospheric test model, in: H. Power, N. Moussiopoulos and C.A. Brebbia, eds., *Air Pollution III, Volume 1: Air Pollution, Theory and Simulation*, (Computational Mechanics Publications, Southampton, Boston, 1995) 319–326.
16. E.J. SPEE, P.M. DE ZEEUW, J.G. VERWER, J.G. BLOM AND W. HUNSDORFER: Vectorization and parallelization of a numerical scheme for 3D global atmospheric transport-chemistry problems, CWI Report NM-R9620, 1996.

17. E.J. SPEE, A.C. PETERSEN, H. VAN DOP AND W. HUNDSDORFER: Sensitivity of atmospheric transport model performance to numerical advection schemes and resolution, CWI Report MAS-97xx, (in preparation) 1997.
18. G. STRANG: On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5 (1968) 506–517.
19. H. THE: Private communication, (RIVM, Bilthoven, 1994).
20. J.G. VERWER: Gauss-Seidel iteration for stiff ODEs from chemical kinetics, *SIAM J. Sci. Statist. Comput.* 15 (1994) 1243–1250.
21. J.G. VERWER AND J.G. BLOM: On the coupled solution of diffusion and chemistry in air pollution models, in: Proceedings of the Third International Congress on Industrial and Applied Mathematics (ICIAM/GAMM 95), *ZAMM*, Issue 4: Applied Sciences, especially Mechanics, eds. Edwin Kreuzer and Oskar Mahrenholtz, Akademie Verlag, pp. 454–457 (1996).
22. J.G. VERWER, J.G. BLOM AND W. HUNDSDORFER: An implicit-explicit approach for atmospheric transport-chemistry problems, *Appl. Numer. Math.* 20 (1996) 191–209.
23. J.G. VERWER, J.G. BLOM, M. VAN LOON AND E.J. SPEE: A comparison of stiff ODE solvers for atmospheric chemistry problems, CWI preprint NM-R9505, *Atmospheric Environment* 30 (1996) 49–58.
24. D.L. WILLIAMSON: Review of numerical approaches for modeling global transport, in: H. van Dop and G. Kallos, eds., *Air Pollution Modeling and its Application IX*, (Plenum Press, New York, 1992).