CWI

Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

Vectorization and parallelization of a numerical scheme for 3D global atmospheric transport-chemistry problems

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

Department of Numerical Mathematics

NM-R9620 1996

Report NM-R9620
ISSN 0169-0388

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

Vectorization and Parallelization of a Numerical Scheme for 3D Global Atmospheric Transport-Chemistry Problems

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

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

Edwin.Spee@cwi.nl, Paul.de.Zeeuw@cwi.nl,
Jan.Verwer@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. In such cases use of the most advanced computer systems is a prerequisite for making real progress. This report is devoted to a vectorization/parallelization study, on a Cray C90, of an efficient numerical scheme in development for global atmospheric transport-chemistry problems. The scheme employs a particular type of operator splitting. Performance results are presented with respect to (grid) vectorization and parallelization based on autotasking and on a division in subdomains of the globe. For this purpose we use a constructed, three-space dimensional model problem containing advection, vertical turbulent diffusion and chemical reactions. We also study the numerical accuracy/efficiency of the operator splitting scheme for this model problem. The model problem is presented as a benchmark on which other schemes and implementations can be tested. The benchmark problem is available through World Wide Web.

AMS Subject Classification (1991): 65M06, 65M20, 65Y05, 65Y20
CR Subject Classification (1991): G.1.7, G.1.8, J.2
Keywords & Phrases: Air quality modeling, global atmospheric transport-chemistry problems, operator splitting, advection schemes, stiff ODE solution, high performance computing, vectorization, parallelization.
Note: This research 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) through the CIRK project.

Table of Contents

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 in fact 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, 11] 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 report is devoted to a vectorization/parallelization study. On a Cray C90 we have implemented an efficient numerical scheme which is in development for solving global atmospheric transport-chemistry problems. The scheme employs a particular type of operator splitting which, like other popular splittings in the field, allows a high degree of vectorization/parallelization. Performance results are presented with respect to (grid) vectorization and parallelization based on autotasking and on a division into subdomains of the globe. The performance results show that our operator splitting scheme vectorizes very well, it easily reaches 500 Mflops on one processor. This is a highly satisfactory speed when taking into account that the theoretical peak on one processor is 952 Mflops. As well-known, however, this peak can only be realized for specific loops allowing chaining of floating point operations. Our study into parallelism was limited to four processors. The results indicate that the speedup from autotasking will be lower than 50% of the optimal speedup. Autotasking takes place at a low algorithmic level so that the start up of small tasks and frequent synchronization between processors induces a considerable amount of overhead. On the other hand, parallelism through division into subdomains takes place at a high algorithmic level and is therefore more efficient. The results indicate that the speedup will easily reach 75% of the optimum and on a sufficiently fine grid it will come close to the maximum predicted by Amdahl's law.

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 also the numerical accuracy/efficiency of the operator splitting scheme. 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. We have made the problem available through World Wide Web [1] 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 our operator splitting technique and the numerical schemes which are used. 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 three parallelization techniques. The final Section 7 is devoted to summarizing conclusions and remarks.

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

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

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

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

$$R(c) = P(c) - L(c)\, c, \tag{2.2}$$

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, 11] 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 $1^o \times 1^o$ 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 height. With the above restrictions in mind, we put $\underline{u} = (u, v)$ and write the divergence in spherical coordinates,

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

Here a is the radius of the earth (6378 km) and we should mention that we have approximated the radial distance factor $1/r$ 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

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

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

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

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

3.1 *Operator splitting*

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. In this paper we also use operator splitting, but we split only once. Following [19], the chemistry and vertical diffusion is kept together and solved in a coupled manner.

We thus rewrite equation (2.1) to

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

where

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

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 $s = 0, 1, \ldots$. Starting from c_s, the approximation c_{s+1} at the next time point t_{s+1} is then computed by solving

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

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

$$\frac{\partial}{\partial t} c^{(3)}(t) = F_2(c^{(3)}(t)) \qquad (t_{s+\frac{1}{2}} \le t \le t_{s+1}), \quad c^{(3)}(t_{s+\frac{1}{2}}) = c^{(2)}(t_{s+1}), \tag{3.7c}$$

and $c_{s+1} \equiv c^{(3)}(t_{s+1})$. Note that the sequence of computations is symmetrical around the time point halfway (Strang splitting [15]).

By means of (3.7), the original problem thus has been separated into two different problems that can be treated with different numerical techniques. Equation (3.7b) will be referred to as the chemistry-diffusion problem and both equation (3.7a) and equation (3.7c) as the advection problem. A difficulty with operator splitting is that even when these subproblems are solved exactly, as is still assumed in (3.7), we still have to face the splitting error. This means that the step size Δt_{split} must be chosen in relation with discretization parameters inside the advection and chemistry-diffusion computation. In general this choice is hard to make and requires experimental insight. In Section 3.4 we return to this issue.

3.2 *The advection computation*

Because there is only horizontal advection, it suffices to describe the advection computation for the 2D equation

$$\frac{\partial c}{\partial t} = -\frac{1}{a \cos\phi}\left[\frac{\partial(u\,c)}{\partial\lambda} + \frac{\partial(v\,c\cos\phi)}{\partial\phi}\right], \tag{3.8}$$

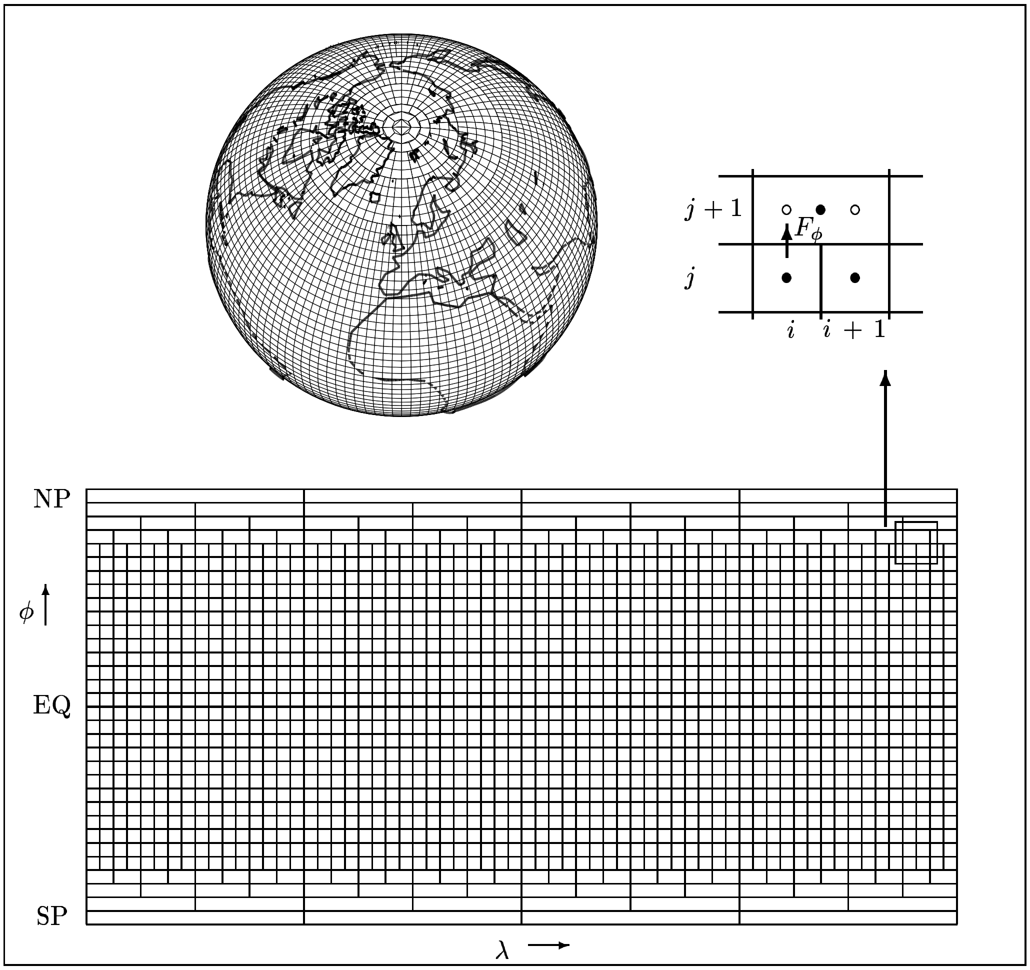


Figure 1: A reduced grid on the globe (left, top), virtual concentrations ∘ and real concentrations • (right, top), and a 64×32 reduced grid in longitude/latitude co-ordinates (bottom).

where $c = c(t, \lambda, \phi)$ is now a scalar rather than a vector. A so-called reduced space grid [21] 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 is detrimental for the stability of explicit advection schemes.

Following the cell-centered approach, the spherical advection operator is 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 (3.8) then reads

$$\frac{d}{dt} c_{i,j} = -\frac{1}{a \cos\phi_j} \left[\frac{\mathrm{fL}_{i+\frac{1}{2},j} - \mathrm{fL}_{i-\frac{1}{2},j}}{\Delta\lambda(j)} + \frac{\mathrm{fP}_{i,j+\frac{1}{2}} - \mathrm{fP}_{i,j-\frac{1}{2}}}{\Delta\phi} \right], \tag{3.9}$$

with the fluxes on the cell boundaries

$$\mathrm{fL}_{i+\frac{1}{2},j} \quad = \quad \max(u_{i+\frac{1}{2},j}\,, 0)\mathrm{fL}^{+}_{i+\frac{1}{2},j} + \min(u_{i+\frac{1}{2},j}\,, 0)\mathrm{fL}^{-}_{i+\frac{1}{2},j}, \tag{3.10a}$$

$$\mathrm{fP}_{i,j+\frac{1}{2}} \;=\; [\max(v_{i,j+\frac{1}{2}}\,,0)\mathrm{fP}^{+}_{i,j+\frac{1}{2}} + \min(v_{i,j+\frac{1}{2}}\,,0)\mathrm{fP}^{-}_{i,j+\frac{1}{2}}]\,\cos(\phi_{j+\frac{1}{2}}). \tag{3.10b}$$

These fluxes are defined by

$$\mathrm{fL}^{+}_{i+\frac{1}{2},j} \;=\; c_{i,j} + \Psi(\theta_{i+\frac{1}{2},j})\cdot(c_{i+1,j}-c_{i,j}), \tag{3.11a}$$

$$\mathrm{fL}^{-}_{i+\frac{1}{2},j} \;=\; c_{i+1,j} + \Psi(\theta^{-1}_{i+\frac{3}{2},j})\cdot(c_{i,j}-c_{i+1,j}), \tag{3.11b}$$

$$\mathrm{fP}^{+}_{i,j+\frac{1}{2}} \;=\; c_{i,j} + \Psi(\theta_{i,j+\frac{1}{2}})\cdot(c_{i,j+1}-c_{i,j}), \tag{3.11c}$$

$$\mathrm{fP}^{-}_{i,j+\frac{1}{2}} \;=\; c_{i,j+1} + \Psi(\theta^{-1}_{i,j+\frac{3}{2}})\cdot(c_{i,j}-c_{i,j+1}), \tag{3.11d}$$

with

$$\theta_{i+\frac{1}{2},j} = \frac{c_{i,j}-c_{i-1,j}}{c_{i+1,j}-c_{i,j}}, \quad \theta_{i,j+\frac{1}{2}} = \frac{c_{i,j}-c_{i,j-1}}{c_{i,j+1}-c_{i,j}}, \tag{3.12}$$

and Ψ is the flux-limiter function [9]

$$\Psi(\theta) = \max(0, \min(1, \theta, \frac{1}{3}+\frac{1}{6}\theta)). \tag{3.13}$$

We used this advection approximation earlier in [1, 5, 14]. In [5] 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.

The use of the reduced grid leads to a few technicalities. 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. Also, where grid reduction takes place, piecewise constant interpolation is used for concentration values whenever needed. For more details we refer to [1] where we developed this reduced grid approach.

At this stage of development, (3.9) 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 (3.9) by the system

$$\frac{d}{dt}\vec{c} = \vec{f}(\vec{c}), \tag{3.14}$$

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 (3.9), here denoted by $f_{i,j}$. The complete advection scheme is then given by

$$\vec{w} \;=\; \vec{c}_m + \Delta t_{\text{adv}}\vec{f}(\vec{c}_m), \tag{3.15a}$$

$$\vec{c}_{m+1} \;=\; \vec{c}_m + \frac{1}{2}\Delta t_{\text{adv}}\left(\vec{f}(\vec{c}_m) + \vec{f}(\vec{w})\right), \tag{3.15b}$$

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 an integer fraction of it.

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 (3.14) by applying flux-limiting. This leads to additional costs for the advection computation. However, in [5] 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 [5]). The results presented there indicate that the combination is stable and positive, as long as Δt_{adv} is adjusted to satisfy the CFL restriction

$$\nu_\lambda + \nu_\phi \;\leq\; 2/3, \tag{3.16}$$

where

$$\nu_\lambda = \max_{i,j} \frac{\Delta t_{\text{adv}}}{\Delta \lambda} \frac{|u(\lambda_{i+\frac{1}{2}}, \phi_j)|}{a \cos \phi_j}, \quad \nu_\phi = \max_{i,j} \frac{\Delta t_{\text{adv}}}{\Delta \phi} \frac{|v(\lambda_i, \phi_{j+\frac{1}{2}})| \cos \phi_{j+\frac{1}{2}}}{a \cos \phi_j}. \tag{3.17}$$

3.3 The chemistry-diffusion computation

While atmospheric advection can be integrated efficiently with an explicit method, atmospheric chemistry gives rise to stiffness, excluding a fully explicit integration. This also holds for vertical turbulent diffusion which transports species as fast as many react with one another. This agreement between time constants suggests to solve the chemistry and vertical diffusion in a coupled manner, rather than to split the two which is usually done. In our case this leads to a computation for the chemistry-diffusion problem

$$\frac{\partial}{\partial t} c = \frac{\partial}{\partial r} \left(\rho K \frac{\partial}{\partial r} (\frac{c}{\rho}) \right) + R(c), \quad 0 \le r \le r_H, \tag{3.18}$$

subjected to the boundary conditions (2.5), 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 [18, 19] 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 [19], Section 2.

It suffices to consider the 1D case so that $c = c(t, r) \in \Re^m$. 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

$$\Omega_V = \{ r_k : r_1 = \frac{1}{2} \Delta r_1, \, r_k = r_{k-1} + \frac{1}{2}(\Delta r_{k-1} + \Delta r_k), \, 2 \le k \le N_r \} \tag{3.19}$$

in the following way,

$$\frac{\partial}{\partial r} \left(\rho K \frac{\partial}{\partial r} (\frac{c}{\rho}) \right) \approx \frac{2}{r_{k+1} - r_{k-1}} \left((\rho K)^+_k \frac{\chi_{k+1} - \chi_k}{r_{k+1} - r_k} - (\rho K)^-_k \frac{\chi_k - \chi_{k-1}}{r_k - r_{k-1}} \right), \tag{3.20}$$

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

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

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,

$$\mathbf{c}^{(j)} = [c_1^{(j)}, \ldots, c_{N_r}^{(j)}]^T, \quad \mathbf{P}^{(j)}(\mathbf{c}) = [P^{(j)}(c_1), \ldots, P^{(j)}(c_{N_r})]^T, \quad j = 1, \ldots, m. \tag{3.21}$$

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

$$\mathbf{L}^{(j)}(\mathbf{c}) = \mathrm{diag}(L^{(j)}(c_1), \ldots, L^{(j)}(c_{N_r})), \quad j = 1, \ldots, m. \tag{3.22}$$

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

$$\frac{d}{dt}\mathbf{c}^{(j)} = \mathbf{A}\mathbf{c}^{(j)} + \mathbf{P}^{(j)}(\mathbf{c}) - \mathbf{L}^{(j)}(\mathbf{c})\mathbf{c}^{(j)}, \quad j = 1, \ldots, m, \tag{3.23}$$

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 [17, 19].

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

$$\mathbf{c}_{n+1}^{(j)} = \mathbf{C}^{(j)} + \gamma \Delta t_{\text{cvd}} \left(\mathbf{A}\mathbf{c}_{n+1}^{(j)} + \mathbf{P}^{(j)}(\mathbf{c}_{n+1}) - \mathbf{L}^{(j)}(\mathbf{c}_{n+1})\mathbf{c}_{n+1}^{(j)} \right), \quad j = 1, \ldots, m, \tag{3.24}$$

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

$$\gamma = \frac{1+q}{1+2q}, \quad \mathbf{C}^{(j)} = \frac{1}{1+2q}\left((1+q)^2 \mathbf{c}_n^{(j)} - q^2 \mathbf{c}_{n-1}^{(j)} \right), \quad q = \frac{t_{n+1} - t_n}{t_n - t_{n-1}}. \tag{3.25}$$

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

The BDF formula (3.24) 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 (3.24) now reads

$$\mathbf{c}^{(j)} = \mathbf{C}^{(j)} + \gamma\tau\mathbf{A}\mathbf{c}^{(j)} + \gamma\tau\mathbf{P}^{(j)}(\mathbf{c}) - \gamma\tau\mathbf{L}^{(j)}(\mathbf{c})\mathbf{c}^{(j)}, \quad j = 1, \ldots, m. \tag{3.26}$$

This system is equivalent to

$$\mathbf{c}^{(j)} = \left(\mathbf{I} - \gamma\tau\mathbf{A} + \gamma\tau\mathbf{L}^{(j)}(\mathbf{c}) \right)^{-1} \left(\mathbf{C}^{(j)} + \gamma\tau\mathbf{P}^{(j)}(\mathbf{c}) \right), \quad j = 1, \ldots, m, \tag{3.27}$$

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 $\mathbf{c}^{(j)}, 1 \le j \le m$, is carried out on equation (3.27) 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 $j = 1, \ldots, m$:
      $2a$. $\mathbf{L}^{(j)}(\mathbf{c}_{[i]}), \mathbf{P}^{(j)}(\mathbf{c}_{[i]})$.
      $2b$. LU-decompose $\mathbf{I} - \gamma\tau\mathbf{A} + \gamma\tau\mathbf{L}^{(j)}(\mathbf{c}_{[i]}))$. (3.28)
      $2c$. 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.

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. Would there be no diffusion, then this process is completely identical to that used for the box models in [17, 20]. On the other hand, without chemistry the diffusion is treated implicitly in the usual way.

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, described later, we in fact used 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 with the usual approach based on modified Newton iteration illustrate this [18, 19].

3.4 Reducing the splitting error

At this stage of development we have discussed the full discretization of the operator splitting scheme (3.7). 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 scheme

$$\frac{\partial}{\partial t}c^{(1)}(t) = F_2(c^{(1)}(t)) \quad (t_s \le t \le t_{s+1}), \quad c^{(1)}(t_s) = c_s, \tag{3.29a}$$

$$\frac{\partial}{\partial t}c^{(2)}(t) = F_1(c^{(2)}(t)) \quad (t_s \le t \le t_{s+1}), \quad c^{(2)}(t_s) = c^{(1)}(t_{s+1}), \tag{3.29b}$$

where $c_{s+1} \equiv c^{(2)}(t_{s+1})$. In [10] and [6] it is shown that for advection-reaction equations the splitting error can be eliminated. Spee [14] has implemented the underlying idea for a 2D advection-reaction problem which is related to the benchmark problem presented in Section 4. In the tests described in the remainder of the paper, for this benchmark problem, we have adopted the implementation of [14] which we present here for reasons of self-containedness.

Assume that the velocities u, v are uniform in the vertical direction (this holds for the benchmark problem) and consider system (3.23) which is integrated in time over the split interval $[t_s, t_{s+1}]$, at any of the cell centers (λ_i, ϕ_j) of the horizontal grid. For convenience of notation we represent it by

$$\frac{d}{dt}\vec{c} = \vec{g}\,(t, \lambda_i, \phi_j, \vec{c}), \tag{3.30}$$

showing now the dependence on t, λ_i, ϕ_j. In (3.23) this dependence was suppressed. Next consider the modified system

$$\frac{d}{dt}\tilde{\vec{c}} = \vec{g}\,(t, \tilde{\lambda}_i(t), \tilde{\phi}_j(t), \tilde{\vec{c}}\,), \tag{3.31}$$

where $\tilde{\lambda}_i(t), \tilde{\phi}_j(t)$ are the characteristics of the horizontal advection operator passing through λ_i, ϕ_j at time $t = t_{s+\frac{1}{2}}$. Solving (3.31) over the interval $[t_s, t_{s+1}]$ would make the advection steps prior to and after the chemistry-diffusion step redundant. This of course would necessitate a moving characteristic grid. Since we use an Eulerian grid approach, a mapping is required from the Eulerian grid to the characteristic grid at time $t = t_s$, and vice versa at time $t = t_{s+1}$. In Eulerian-Lagrangian methods this is achieved by an interpolation procedure. In our splitting method the advection steps prior to and after the chemistry-diffusion step can be interpreted this way. In other words, when solving (3.31), the only errors we make are the usual discretization errors in the advection and chemistry-diffusion steps, showing that the splitting error has been completely eliminated. From this derivation we also see that if the right-hand side function $\vec{g}$ is independent of λ, ϕ, the splitting error does not exist and hence no modification of (3.30) is required. This of course will seldom be the case.

The characteristics are solutions of the differential equations

$$\dot{\lambda} = \frac{u}{a\,\cos\phi}, \quad \dot{\phi} = \frac{v}{a}. \tag{3.32}$$

Rather than numerically integrate (3.32) to high accuracy, we use a tangent approximation in (3.31). Thus, in the chemistry-diffusion computation we replace (3.30) by the modified system

$$\frac{d}{dt}\vec{c} = \vec{g}\,(t, \lambda_i + \frac{u}{a\,\cos\phi_i}\,(t - t_{s+\frac{1}{2}}), \phi_j + \frac{v}{a}\,(t - t_{s+\frac{1}{2}}), \vec{c}), \tag{3.33}$$

where u, v are the velocities at the point (λ_i, ϕ_j) at the intermediate time $t_{s+\frac{1}{2}}$.

In conclusion, rather than truly eliminating the splitting error, we reduce it. As said above, for this splitting error reduction to work, the velocities u, v need to be uniform in the vertical direction. Needless to say that in the actual practice this renders a serious restriction. Nevertheless it is interesting to see that in this special restricted case the technique pays off. We will illustrate this in Section 5.3.

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

For measuring vectorization/parallelization performances, and to test the numerical schemes, we have developed a benchmark problem for 3D global transport and chemistry. This benchmark problem is similar to the regional test problem used in [19]. When we omit the vertical diffusion, the problem is very similar to the 2D one used in [14]. The complete description and a reference solution are available through World Wide Web [2]. 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 and hence 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 [16]. This model is fully described in the appendix of the preprint to [20]. 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). 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 5 days, starting at midnight Greenwich Mean Time. In applications much longer time spans are seen, from months to even a year say. We choose a relatively short time interval in order not to consume a too large part of our CPU budget. Five days is certainly long enough for measuring vectorization/parallelization performances. With the exception of HNO_3 and NO, for all species the initial concentrations are at a uniform background level over the globe [3]. 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 shaped as depicted in Figure 2. We emphasize that a cylinder shaped initial condition is a challenge for any advection scheme. The center of the cylinder lies at (λ', ϕ' = 12.6^{o}E, 17.712^{o}S). This location is chosen such that at the end of the integration time of five days the center of

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

[3] Hence the initial concentrations are far from the diurnal photochemical equilibrium. This will cause strong initial transients at the start of the integration.

the cylinder lies on the line $\phi' = \lambda'/2$. 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 initial concentrations in the other vertical levels are such that in each vertical column the mixing ratio is constant.

	Name	Species	Concentration
	-	M	$2.55 * 10^{19}$
	Water	H_2O	$2.55 * 10^{17}$
	Carbon monoxide	CO	$2.55 * 10^{12}$
	Molecular oxigen	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 oxigen	O^1D	0.0
17	Atomic oxigen (g.s.)	O^3P	0.0

Table 1: Initial concentrations at ground level in [molec/cm^3].

4.3 Advection

A divergence free, horizontal wind field in analytical form is used to enable the computation of an exact solution along characteristics, similar as in [14, 19]. Hence vertical advection does not take place. The wind field stems from [13] and describes a solid body rotation, see Figure 3. The velocities are given by

$$u = 2\pi\,\kappa\,(\cos\beta\cos\phi + \sin\beta\sin\phi\cos\lambda), \quad v = -2\pi\,\kappa\,\sin\beta\sin\lambda, \tag{4.34}$$

where $\beta = 45^o$ and $\kappa = a/(14 \times 24 \times 3600)$. Observe that the wind velocities are constant in the vertical direction and in time. The maximal wind speed is approximately 119 km/hours. Near the surface this is of course very strong, but higher up in the atmosphere these velocities do occur.

4.4 Vertical diffusion

The vertical turbulent diffusion coefficient K depends on the mixing height which depends on the time of day. The same constructed expression as in [19] is used. At night $K \approx 0$, while shortly after noon it reaches its maximal value. We have used 15 cell-centered vertical layers to approximate the vertical diffusion term. The distribution of the cell centers is a function of the pressure which is taken uniform over the globe. The lowest cell center lies at sea level (1000 hPa) and the highest at 22.6 km (0 hPa). The complete distribution of the cell centers 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.

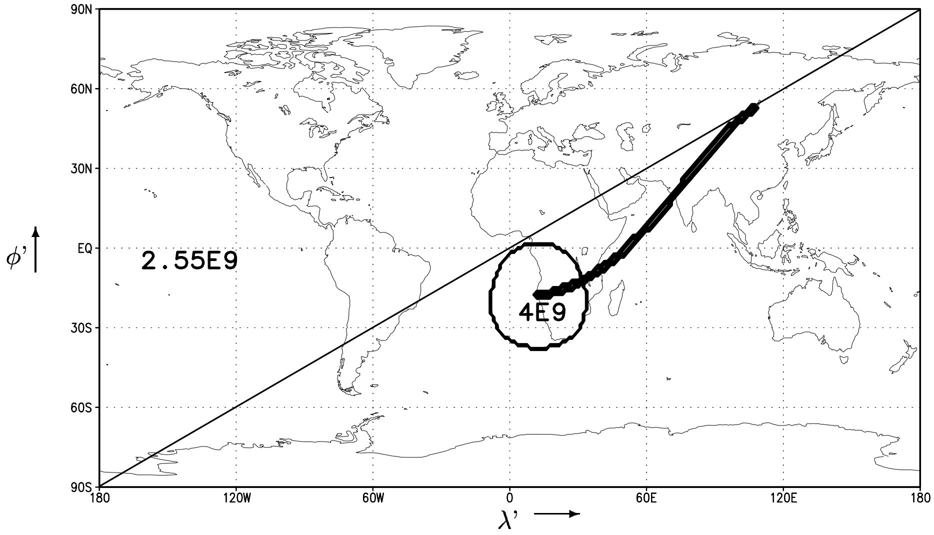


Figure 2: The initial concentration vector for HNO_3 and the trajectory of the cylinder. The coordinates are in degrees: λ'$= (\lambda - \pi)\frac{180}{\pi}$ and ϕ'$=\phi\frac{180}{\pi}$.

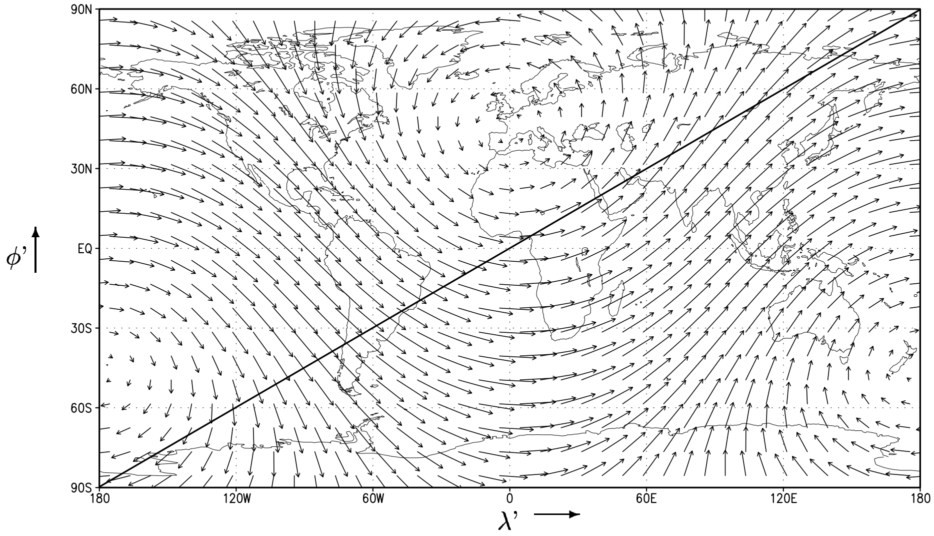


Figure 3: The prescribed wind-field (λ'$=(\lambda - \pi)\frac{180}{\pi}$; ϕ'$=\phi\frac{180}{\pi}$).

4.5 The reference solution

Suppose, for a moment, that there is no vertical diffusion and chemistry. Exactly after five days, the cylindrically shaped initial concentration profiles for HNO_3 and NO then would be centered at the diagonal line $\lambda' = \phi'/2$ at the point indicated in Figure 2. We therefore have computed an accurate reference solution at all vertical layers lying above this 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 [19]. Hence our reference solution can be considered exact except for the error due to the discretization of the vertical diffusion term. This means that assessment of accuracy is possible only for the horizontal advective transport and the solution of the chemistry. Figure 4 shows the reference solution for HNO_2 and O_3 along the diagonal line at ground level. For O_3 the cylindrical profile imposed for HNO_3 and NO is clearly visible. The HNO_2 profile shows that a rather fine grid is needed to resolve all details. Observe that at the initial time the concentration for HNO_2 and O_3 was uniformly distributed at background level.

The semi-discrete reference solution has been computed at 128 fixed positions along the diagonal line for all species. For further use we define a relative error Err, based on this specific reference solution set. The main purpose of Err is to provide a quantity that can be used in the comparison of solvers. Err is defined by

$$Err = \left(\sqrt{\sum_{i=1}^{128} \sum_{k=1}^{15} \cos\phi_i (y_{i,k} - \hat{y}_{i,k})^2 \Big/ \sum_{i=1}^{128} \sum_{k=1}^{15} \cos\phi_i} \right) \Big/ \max_{i,k} \hat{y}_{i,k}. \tag{4.35}$$

The variable $\hat{y}_{i,k}$ represents for a selected species the reference solution value in layer k at the i-th location on the diagonal line. The variable $y_{i,k}$ represents its computed counterpart, which is supposed to be obtained by interpolating values from the space grid in use. The cosines in the error expression serve to weight the errors such that those near the poles get a small weight compared to those in the neighbourhood of the equator.

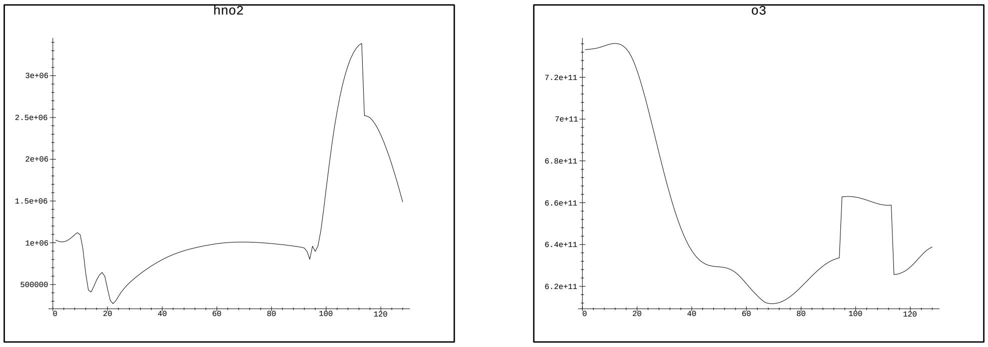


Figure 4: Reference solution along the diagonal line at ground level, HNO_2 (left) and O_3 (right).

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 neglected 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 five days is short and that our chemical scheme composed of 17 species is of moderate size, see [11]. The same can be said about the space grids. The finest longitude/latitude grid here is approximately $1.4^o \times 1.4^o$. In [11] it is pointed out that still finer grids are needed in future generation models.

Table 2: Problem size.

Grid (λ, ϕ, r)	# grid cells	# unknowns
64 × 32 × 15	30.720	522.240
128 × 64 × 15	122.880	2.088.960
256 × 128 × 15	491.520	8.355.840

5. Vectorization, performance, accuracy and efficiency

In this section we outline how the splitting method is vectorized, we present results of performance tests on the benchmark problem and comment on the numerical accuracy and efficiency observed in these tests.

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 is similar to that in [19] 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_{cvd}. 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 far from 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 often 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 [19] clustering has not been considered.

It is 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. By increasing the vector length, overhead costs become relatively smaller. 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 standard FORTRAN 77, on the 4-processor Cray-C98/4256 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, 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 in our runs were chosen as follows. In all cases the operator split step size $\Delta t_{\text{split}} = 14400$ sec. Convergence checks for the advection and chemistry-diffusion computation have revealed that for the current benchmark problem four hours is sufficiently small for the accuracy level required. We owe this to the use of the characteristics in the chemistry-diffusion step. See Section 5.3 for an illustration of this remark. The advection step size Δt_{adv} is related to $\Delta\lambda$ and $\Delta\phi$ in connection with the CFL condition (3.16) for stability and positivity. The actual advection step sizes Δt_{adv} used were taken slightly smaller to form an integer fraction of half the split step size. They are given in the tables of result. The stepsize Δt_{cvd} for the chemistry-diffusion computation varies in time and is governed by the step size control mechanism of the integration method. It is constrained by an imposed minimum $\Delta t_{\text{cvd}}^{\text{min}}$ and a maximum, the latter always being 5 times as large as the minimum. The choice for this minimum step size determines to a large extent the CPU time for the chemistry-diffusion computation.

Table 3 shows performance results for two values of $\Delta t_{\text{cvd}}^{\text{min}}$ and three different longitude-latitude grids. The value for Δt_{cvd} given in this table is the average value. The performance in terms of CPU times and Mflop rates is shown for the advection implementation, the chemistry-diffusion implementation and the implementation as a whole. For the implementation as a whole we observe very satisfactory Mflop rates, close to 500 throughout. On the coarsest grid the Mflop rate in the advection computation is low due to the grid reduction. The finer the grid, the smaller will the influence of grid reduction be. The advection part starts to outweigh the chemistry-diffusion part for increasing grid dimensions, due to diminishing Δt_{adv} for preservation of the CFL condition. Moreover, for increasing grid dimensions, the clusters used in the chemistry-diffusion computation will cover a smaller area on the globe which means that the step sizes Δt_{cvd} can grow. This also makes the chemistry-diffusion computation less expensive. The table reveals the growth in the step size Δt_{cvd} clearly when going from the coarsest to the second coarsest grid.

5.3 Accuracy

We examine the accuracy in two ways. First, for the six tests tabulated in Table 3, Figures 5 and 6 show plots of the computed and reference solution concentrations of HNO_2 and O_3, in the same way the reference solution was depicted in Figure 4. Going to a finer grid clearly resolves all details in the solutions, including the real wiggles in HNO_2 and those parts of the cylinder shaped initial profile still present after five days. Going to a more accurate chemistry-diffusion computation further increases the accuracy, but it should be emphasized that the computations for a minimum step size of 360 sec. are already sufficiently accurate for the actual practice. The practical importance of this observation becomes evident when we take the CPU times given in Table 3 into account. The most expensive, finest grid computation with a minimum of 60 sec. (case b3), is clearly overdone with respect to accuracy needed in the actual practice. On the other hand, the cheapest, coarsest grid computation with a minimum of 360 sec. (case a1) is already sufficiently accurate for many practical purposes where fine solution details are less important. Observe that the CPU times for these two

Table 3: Performance on one processor of the Cray-C98/4256 at SARA, Amsterdam.

(a) Minimal chemistry-diffusion step size $\Delta t_{\rm cvd}^{\rm min} = 360$.

Grid (λ, ϕ, r)	Advection			Chemistry-diffusion			Total	
	$\Delta t_{\rm adv}$	CPU	rate	$\Delta t_{\rm cvd}$	CPU	rate	rate	CPU
	(secs.)	%	(Mflops)	(secs.)	%	(Mflops)	(Mflops)	(secs.)
64 × 32 × 15	2400	31	387	360	65	499	446	171
128 × 64 × 15	1200	44	485	426	54	491	479	698
256 × 128 × 15	600	58	548	427	41	483	515	3870

(b) Minimal chemistry-diffusion step size $\Delta t_{\rm cvd}^{\rm min} = 60$.

Grid (λ, ϕ, r)	Advection			Chemistry-diffusion			Total	
	$\Delta t_{\rm adv}$	CPU	rate	$\Delta t_{\rm cvd}$	CPU	rate	rate	CPU
	(secs.)	%	(Mflops)	(secs.)	%	(Mflops)	(Mflops)	(secs.)
64 × 32 × 15	2400	8	384	67	91	493	480	664
128 × 64 × 15	1200	15	485	90	84	491	487	2040
256 × 128 × 15	600	23	550	88	76	487	500	9590

Table 4: Relative errors (4.35) for the six tests tabulated in Table 3.

(a) Minimal chemistry-diffusion step size $\Delta t_{\rm cvd}^{\rm min} = 360$.

Grid (λ, ϕ, r)	$\Delta t_{\rm adv}$	$\Delta t_{\rm cvd}$	HNO_2	O_3	max.
64 × 32 × 15	2400	360	1.3E − 2	3.3E − 3	2.4E − 2
128 × 64 × 15	1200	426	1.0E − 2	3.1E − 3	2.0E − 2
256 × 128 × 15	600	427	8.7E − 3	2.8E − 3	1.8E − 2

(b) Minimal chemistry-diffusion step size $\Delta t_{\rm cvd}^{\rm min} = 60$.

Grid (λ, ϕ, r)	$\Delta t_{\rm adv}$	$\Delta t_{\rm cvd}$	HNO_2	O_3	max.
64 × 32 × 15	2400	67	1.1E − 2	2.4E − 3	2.1E − 2
128 × 64 × 15	1200	90	8.5E − 3	1.8E − 3	1.6E − 2
256 × 128 × 15	600	88	5.7E − 3	1.3E − 3	1.4E − 2

cases are, respectively, 9590 sec. and 171 sec.

In addition to the figures, we give in Table 4 the relative error (4.35) for HNO_2 and O_3, as well as the maximum taken over all species. Like the figures, the table shows that going to a finer grid and using a smaller minimum for $\Delta t_{\rm cvd}$, results in a higher accuracy. The decrease in these relative errors is rather minor, though. This is due to the fact that very rapid solution transitions in space and time occur. These rapid transitions would require still more accurate computations to better see convergence in the figures of the table.

To illustrate the effect on the accuracy of using the characteristics in the chemistry-diffusion computation, we have included Figure 7. This figure shows for HNO_2, case (a3), the reference solution and the computed solution with and without the use of the characteristics. It is obvious that the use of the characteristics increases the accuracy. We should emphasize, however, that $\Delta t_{\rm split} = 4$ hours, which is very large. Normally one takes $\Delta t_{\rm split} = 2\Delta t_{\rm adv}$, which in this case means a factor of 12 smaller. An additional experiment has indeed shown that with this smaller $\Delta t_{\rm split}$, the splitting error is no longer visible in the HNO_2 solution, see also Figure 7. In this additional experiment, the CPU increases from 3870 sec. to 5130 sec., which is due to a more expensive chemistry-diffusion computation. Since we use variable step sizes, and always restart with the minimum step size allowed, this computation is more efficient on longer time intervals.

5.4 Efficiency

Efficiency is more difficult to assess since we cannot compare our results with results obtained with other solvers. However, the following observation indicates that our algorithm and its vectorized implementation are quite efficient. The upper part (a) of Table 3 shows that the CPU time for the advection computation on the finer grids is larger than for the chemistry and vertical diffusion together. This means that the coupled chemistry-diffusion computation works efficiently when taking the advection computation as a reference point. Of course, the accuracy level of the tests of the upper table (a) 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 [11]. 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 $1.4^o \times 1.4^o \times 15$ grid computation, upper table (a), we have a ratio of about 100:1 against 500 Mflops. Comparing this ratio with the ones given in [11], 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 [11] is 20 to 30 Gflops, whereas we achieve 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 [11].

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. Three different kinds of parallelization will be considered, autotasking, parallelization over the geometry and parallelization over the chemistry.

6.1 Parallelization by autotasking

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 the elementary linear algebra 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 is in nested do-loops (without calls to subroutines). 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. For the advection computation we can rely on autotasking, because the explicit Runge-Kutta method invokes merely simple linear algebra operations on long vectors. Autotasking of the chemistry-diffusion problem is less efficient due to relatively high overhead costs. Most of the overhead comes from entering and leaving parallel regions.

Suppose we have a number of p processors available for parallel execution of a code. We define the speedup S by $S = T(1)/T(p)$, where $T(p)$ is the wall-clock time required to execute the code on p processors. Regarding our expectations, we then have to bear in mind the restriction put by Amdahl's law, which predicts

$$S = \frac{1}{(1-f) + f/p}, \tag{6.36}$$

where f denotes the fraction of the work that can be executed in parallel. In our case, a maximum of four processors was available. This implies that for a speedup $S \geq 2$ we need $f \geq \frac{2}{3}$ and $f \geq \frac{8}{9}$ for a speedup $S \geq 3$. This concerns the ideal case of no costs of overhead.

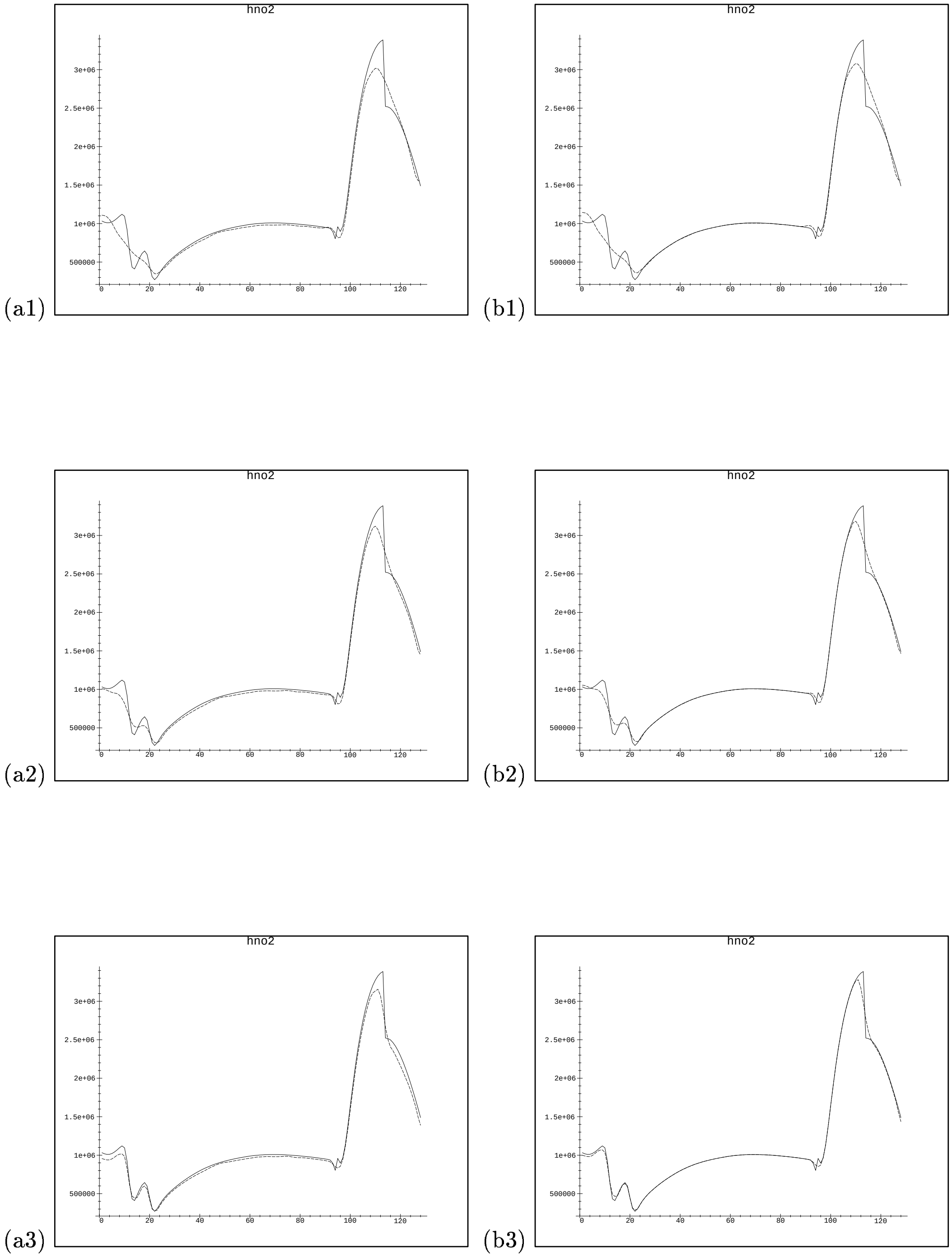


Figure 5: Reference solution and computed solution of HNO_2, depicted along the diagonal line as in Fig. 4, for the six test cases tabulated in Table 3. In the left column $\Delta t_{\mathrm{cvd}}^{\mathrm{min}} = 360$, in the right column $\Delta t_{\mathrm{cvd}}^{\mathrm{min}} = 60$. The two upper plots are for the coarsest grid, etc.

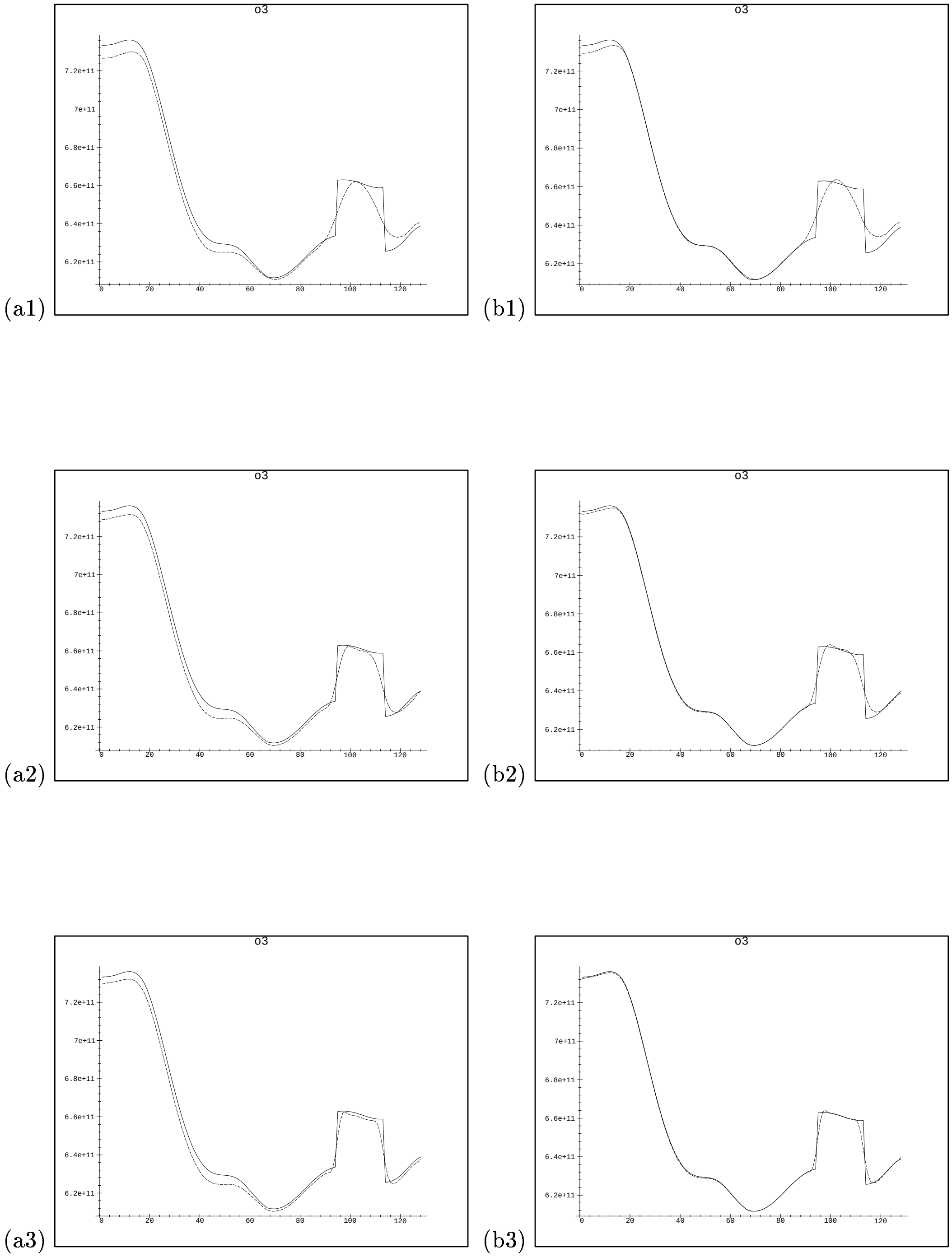
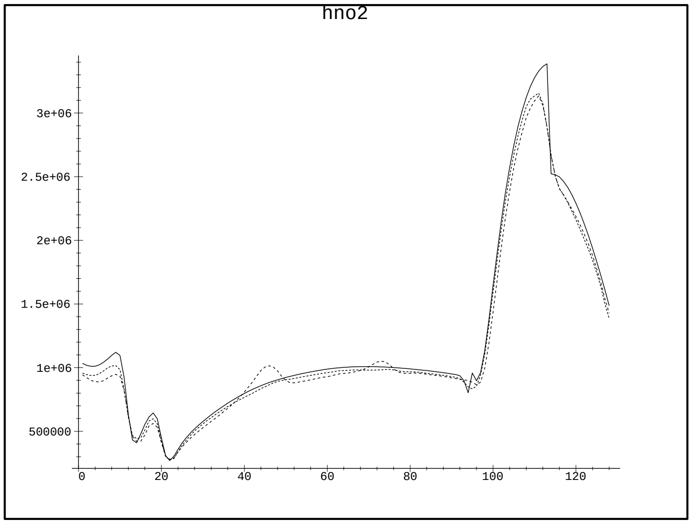


Figure 6: Reference solution and computed solution of O_3, depicted along the diagonal line as in Fig. 4, for the six test cases tabulated in Table 3. In the left column $\Delta t_{\mathrm{cvd}}^{\mathrm{min}} = 360$, in the right column $\Delta t_{\mathrm{cvd}}^{\mathrm{min}} = 60$. The two upper plots are for the coarsest grid, etc.



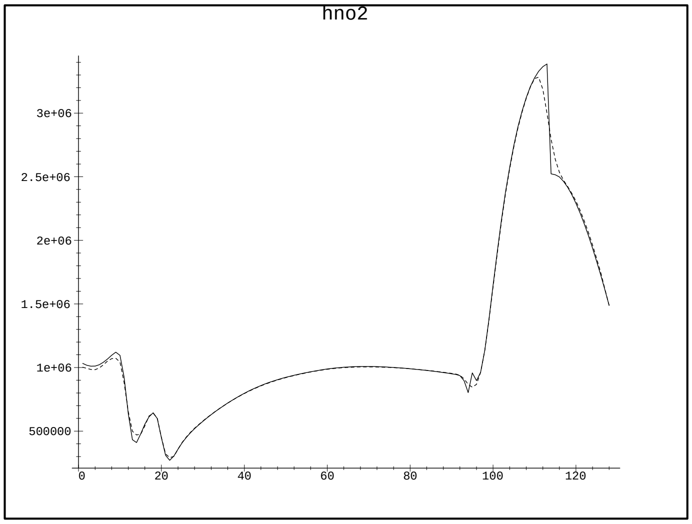


Figure 7: Reference solution (solid line) and computed solutions of HNO_2, case (a3). In the upper plot solutions are presented which were computed with and without the use of the characteristics. The solution with the wiggles is the one computed without the characteristics. The lower plot also shows a solution computed without the characteristics, but now with $\Delta t_{\text{split}} = 2\Delta t_{\text{adv}} = 1200$ sec. instead of 4 hours.

The effects of parallelization can be measured by means of the Cray tool `atexpert`. This tool predicts speedups on a dedicated machine from data collected from a run on a non-dedicated machine [3]. Table 5 contains the predicted values for the six test cases of Table 3, assuming four processors. The bracketed numbers correspond to the ideal speedup according to Amdahl's law (6.36). We measured different speedup factors for the six test cases and all are disappointing. Part (b) of the table clearly reveals that the chemistry-diffusion computation parallelizes badly by autotasking.

6.2 Parallelization over the geometry

As already observed in Section 6.1, parallelization by autotasking suffices for the advection part of the problem. For the chemistry-diffusion part we here consider 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. We used so-called (Cray) CMIC-directives 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_{cvd} 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. Speedup factors are again shown in Table 5. Apparently, the advection and chemistry-diffusion computation are now equally well parallelized and much better than with autotasking.

Table 5: Speedup factors for 4 processors.

(a) Minimal chemistry-diffusion step size $\Delta t_{\text{cvd}}^{\text{min}} = 360$.

$Grid(\lambda, \phi, r)$	Δt_{adv}	Δt_{cvd}	Parall. by autotasking	Parall. over geometry	Parall. over chemistry
64 × 32 × 15	2400	360	1.5 (1.7)	2.2 (2.4)	1.6 (1.7)
128 × 64 × 15	1200	426	1.9 (2.2)	3.1 (3.4)	2.0 (2.2)
256 × 128 × 15	600	427	2.2 (2.4)	3.2 (3.8)	2.4 (2.7)

(b) Minimal chemistry-diffusion step size $\Delta t_{\text{cvd}}^{\text{min}} = 60$.

$Grid(\lambda, \phi, r)$	Δt_{adv}	Δt_{cvd}	Parall. by autotasking	Parall. over geometry	Parall. over chemistry
64 × 32 × 15	2400	67	1.5 (1.7)	2.7 (3.3)	1.6 (1.8)
128 × 64 × 15	1200	90	1.6 (1.7)	3.3 (3.6)	1.8 (2.2)
256 × 128 × 15	600	88	1.6 (1.7)	3.1 (3.8)	2.1 (2.8)

6.3 Parallelization over the chemistry

For the chemistry-diffusion computation we have considered still another type of parallelization, one that changes the Gauss-Seidel algorithm (for the advection part we again relied on autotasking). We have turned the Gauss-Seidel iteration into an iteration which is a mixture between Gauss-Seidel and Jacobi iteration. Hereby, at each processor a Gauss-Seidel iteration is applied for a subset of species. This iteration is done separately from, and in parallel with, similar Gauss-Seidel iterations for species in other subsets on other processors. Observed speedup factors are again given in Table 5. The factors are significantly smaller than for parallelizing over the geometry, though somewhat better than with autotasking. We therefore refrain from further discussing this technique in greater detail. In theory it is possible to combine parallelization over the geometry with parallelization over the chemistry. This possibility has not been investigated as it would lead to a rather complicated implementation.

7. Conclusions and final remarks

Air quality modeling on a global scale is numerically expensive and requires fast algorithms and sophisticated numerical software on high-speed computers. Using a powerful, shared memory, vector/parallel computer, a Cray C90 with 4 processors, we have demonstrated an efficient operator splitting scheme for a test 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 model problems, comprehensive testing of solvers is hardly possible, let alone comparison and validation of solvers.

Our operator splitting scheme is special in that it solves the stiff chemistry and the vertical diffusion in a coupled manner. We have paid a lot of attention to how to vectorize and parallelize this coupled computation on the C90. We have shown that vectorization along the horizontal grid dimension is close to optimal. The Mflop rate measured is about 500. On fine grids, vectorization of the explicit horizontal advection scheme, which makes use of a reduced grid near the poles to alleviate the stability restriction, also leads to about 500 Mflops. These high flop rates were measured on one processor. We also examined three parallelization techniques. One of these, of a domain decomposition type, shows very good speedups and is recommended for the actual practice.

This work will be continued in the near future into two directions. First we will examine an alternative operator splitting approach, where the chemistry and vertical diffusion will be treated uncoupled as well. Our main motive is to test the recently proposed box-model approach of [12] for the solution of the stiff chemistry, in the setting of 3D transport and chemistry. The current benchmark will be used. Second, the work will be extended towards developing and using a second benchmark problem, now involving real meteorological data and orography. The main purpose here is to further test our advection scheme under realistic meteorology and orography conditions. The influence massive I/O operations can have on the program performance will also be examined.

References

1. 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).
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. W. Hundsdorfer, B. Koren, M. van Loon and J.G. Verwer: A positive finite-difference advection scheme, *J. Comput. Phys.* 117 (1995) 35–46.
6. W. Hundsdorfer and J.G. Verwer: A note on splitting errors for advection-reaction equations, *Appl. Numer. Math.* 18 (1995) 191–199.
7. W. Hundsdorfer and E.J. Spee: An efficient horizontal advection scheme for modelling of global transport of constituents, *Monthly Weather Review* 123 (1995) 3554–3564.
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. 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.

10. 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.

11. 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.

12. 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).

13. 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.

14. 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.

15. G. STRANG: On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5 (1968) 506–517.

16. H. THE: Private communication, (RIVM, Bilthoven, 1994).

17. J.G. VERWER: Gauss-Seidel iteration for stiff ODEs from chemical kinetics, *SIAM J. Sci. Statist. Comput.* 15 (1994) 1243–1250.

18. 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).

19. 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.

20. 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.

21. 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).