Surv. Math. Ind. (2002) 10: 107–174

Surveys on Mathematics for Industry

Numerical time integration for air pollution models *

Jan G. Verwer, Willem H. Hundsdorfer, and **Joke G. Blom,** Amsterdam

Summary. Due to the large number of chemical species and the three space dimensions, off-the-shelf stiff ordinary-differential-equation integrators are not feasible for the numerical time integration of stiff systems of advection-diffusion-reaction equations from the field of air pollution modelling. This has led to the use of special time integration techniques. This paper is devoted to a survey of such techniques, encompassing stiff chemistry solvers, positive advection schemes, time or operator splitting, implicit-explicit methods, and approximate matrix factorization solutions. Of great importance in practice is high-performance computing due to the huge problem scales, in particular for global models. We therefore also report on experiences with vector/parallel shared-memory and massively parallel distributed-memory architectures and clusters of workstations. The survey is not entirely unique to air pollution models and biased towards work done at the Centrum voor Wiskunde en Informatica, Amsterdam.

AMS subject classification: 65M06, 65M20, 65Y05, 65Y20.

Keywords: Numerical time integration; Air pollution problem; Reactive flow problem.

1 Introduction

The subject of this paper concerns the numerical solution of advection-diffusion-reaction systems

$$\frac{\partial c}{\partial t} + \nabla \cdot (\underline{u} c) = \nabla \cdot (\underline{K} \nabla c) + R(c), \quad c = c(\underline{x}, t), \quad c \in \mathbf{R}^m, \quad \underline{x} \in \Omega \subset \mathbf{R}^3. \quad (1)$$

Although our findings do have a wider scope, the motivating application is atmospheric-air-quality modelling, where partial-differential-equation (PDE) systems like (1) lie at the heart of complicated models employed in studies on the chemical composition of the atmosphere. The societal motivation for these studies concerns air pollution [32, 116]. The wind field $\underline{u}$ and the diffusion coefficient matrix $\underline{K}$ are given, so that the problem is linear with respect to the transport part; c represents a vector of m concentrations of trace gases like ozone, methane, nitrogen oxides, hydrocarbons, and radicals. The chemical reactions between these species are of first

* Invited paper presented at the International Conference on Air Pollution Modelling and Simulation APMS '98, organized by ENPC and INRIA, October 26–29, 1998, Champs-sur-Marne, France.

and second order, so that the reaction term R is quadratically nonlinear. This term introduces stiffness into the problem as the range of characteristic reaction times in the atmosphere is very large. Emissions and depositions can also be contained in R. Without the chemistry term R the m equations of (1) are uncoupled. The coefficients $\underline{u}$ and $\underline{K}$ and the reaction term R are allowed to depend on the spatial variable $\underline{x}$ and time t. As a rule, this dependence is suppressed in our notation. Boundary conditions for (1) will be specified only when explicitly needed.

Solution methods often used in air pollution modelling follow the method of lines (MOL) approach, i.e., the PDE system with its boundary conditions is first discretized in space on a three-dimensional Eulerian grid, resulting in a huge, continuous-time ordinary-differential-equation (ODE) system. On this grid this system is then solved in time by a numerical integration technique. Due to the huge ODE dimensions, off-the-shelf stiff ODE integrators based on standard numerical algebra routines are not feasible. This has led to the use of special time integration techniques. This paper is devoted to a survey of such techniques, encompassing time or operator splitting, implicit-explicit (IMEX) methods, and approximate matrix factorization solutions. The typical nature of the problem has also led to special stiff chemistry solvers and positive advection schemes. These solvers and schemes are in fact the work horses in operational three-dimensional (3-D) transport-chemistry solvers and consume most of the CPU time. Therefore, a number of chemistry solvers and advection schemes will be reviewed. One of the better advection schemes we discuss does not fit in the MOL approach, as it is based on a direct space-time discretization. Use of this direct scheme in solving (1) presupposes operator or time splitting.

Because in many large-scale applications computer capacity is still a limiting factor for resolution, attention will be paid to high-performance computing. We will report on experiences with vector/parallel shared-memory and massively parallel distributed-memory architectures and clusters of workstations. The survey is not entirely unique to air pollution models and biased towards work done at the Centrum voor Wiskunde en Informatica, Amsterdam.

2 Time integration concerns

General numerical time integration concerns today in the field of air pollution modelling are classic ones – efficiency, stability, accuracy, positivity, mass conservation, noniterative schemes, vectorization, and parallelism –, but as yet their importance is still undiminished.

Efficiency is self-evident, but we should keep in mind that in spite of the tremendous increase in computer speed during the last years, computational speed still is one of the main concerns in the field. For example, global scenario studies easily require many hours of CPU time on the fast supercomputers of today.

Stability is also self-evident, but particularly important for advection and the stiff chemistry solution which easily accounts for over 50% of the total CPU time.

Stiff solvers should remain stable for large step sizes and not be sensitive to initial transients which are present all the time when operator splitting is used.

In air pollution modelling, one is interested in extremely fast, lowly accurate solutions. High accuracy is superfluous since most of the data is not so accurate. As a rule of thumb, approximately 1% relative accuracy for important output species is considered sufficient. Generating lowly accurate solutions at minimal costs and large step sizes is numerically delicate as one then may operate too far from asymptotic assumptions on which methods are based.

Keeping the concentration values positive is not only natural but also necessary for a stable chemistry solution. Hence the requirement of positivity is one of the major concerns. It is encountered mainly in the application of stiff chemistry solvers and numerical advection schemes, for the latter in both the spatial discretization and time integration parts. Positivity for advection schemes is intimately related with monotonicity. As a rule, higher-order schemes do not a priori maintain positivity.

Since the models are in fact mass balances, mass conservation is natural. Standard stiff ODE solvers from the Runge–Kutta and linear multistep families mimic mass conservation for the chemical-kinetics system. The popular special-purpose chemistry solvers QSSA [37], EBI [36], and Twostep [100] do not, as these solvers do not operate with the analytic Jacobian matrix $R'(c)$. Hence these methods need special attention in connection with mass conservation. In the spatial discretization of the advection and diffusion operator, mass conservation is trivially maintained if the flux form is used.

Integration methods for stiff chemistry problems often employ a form of iteration and different temporal step sizes over the space grid. This may create load-balancing problems when multiple processors are used, as, e.g., exemplified in Elbern [25, 26]. "Noniterative" methods, requiring a fixed amount of work and allowing one and the same temporal step size over the grid, are therefore of interest in connection with parallelism. The linearly implicit Rosenbrock method proposed in ref. [106] is of noniterative type as it avoids (Newton) iteration. This method has been successfully tested with equal temporal step size over the grid.

For efficient use of vector computers and cache memory, "grid vectorization" is important. This technique is helpful if at many grid points identical, uncoupled computations need to be done which themselves do not vectorize, like the LU decomposition. By reversing loops, vectorization over the grid then becomes efficient. Grid vectorization therefore should be considered in the selection of algorithms. Fortunately, many algorithms for PDEs lend themselves for grid vectorization. Performance results for air pollution models have been reported in refs. [44], [87] and [104]. As far as we know, grid vectorization, or vectorization across linear systems, was first mentioned in ref. [31].

PDEs of type (1) also lend themselves very well for parallel computing. Using domain decomposition and the communication interface MPI, portable scalable code can be developed for vector/parallel shared-memory and massively parallel distributed-memory architectures and clusters of workstations. However, for reac-

tive flow problems with many reactions and species, memory use is more or less random. Consequently, processors with a hierarchical type of memory (registers, primary/secondary cache, memory, disk) show as yet a rather disappointing flop rate performance. Illustrative experiences can be found in ref. [13].

Two specific subjects of interest not discussed in this paper are adaptivity and reduction.

For complicated PDE calculations in air pollution modelling, grid adaptivity is of interest. With grid adaptivity one attempts to reduce the number of grid points without losing overall accuracy. For example, van Loon [63] has implemented local refinement in a smog prediction model, using a uniform-grid-based cascade technique producing nested grids within one basic time step. Tomlin et al. [98] refined a powerful unstructured triangular grid to solve 2-D problems with spatially inhomogeneous source terms.

Reducing techniques are of interest for large-scale reactive flow problems. The idea is to somehow dynamically "remove" chemical species which live on a "fast time scale", under the assumption that they are "insignificant" for the "slow time scale" or "smooth manifold" solution one is normally interested in. This idea is central, for example, in singular perturbation theory applied to stiff ODEs. Research into reduction in combustion modelling [64] has renewed interest in this subject. See also refs. [22], [90], and [92] and the references therein.

3 The chemical kinetics system

3.1 General observations

Atmospheric chemical-kinetics systems describe chemical reactions between trace gases, such as ozone, nitrogen oxides, methane, and hydrocarbons. One often studies ozone levels in the lower troposphere, as ozone is dangerous for humans and animals during short-term smog episodes and can damage crops when over longer seasonal periods its levels are too high. Ozone is also a greenhouse gas, similar as methane, carbon dioxide, and other species. Air pollution models are therefore also used in connection with climate studies. Ozone itself is not emitted but formed in very many different reactions. A nice introduction to the field of atmospheric chemistry is given by Graedel and Crutzen [32].

Example. As a simple illustration, we consider the following reactions which are basic to any tropospheric air pollution model:

$$\begin{aligned} \mathrm{NO_2} + h\nu &\xrightarrow{k_1} \mathrm{NO} + \mathrm{O}(^3\mathrm{P}) \ , \\ \mathrm{O}(^3\mathrm{P}) + \mathrm{O_2} &\xrightarrow{k_2} \mathrm{O_3} \ , \\ \mathrm{NO} + \mathrm{O_3} &\xrightarrow{k_3} \mathrm{O_2} + \mathrm{NO_2} \ . \end{aligned} \tag{2}$$

Putting $c_1 = [O(^3P)]$, $c_2 = [NO]$, $c_3 = [NO_2]$, and $c_4 = [O_3]$, the associated ODE system reads

$$\begin{aligned}
\dot{c}_1 &= k_1 c_3 - k_2 c_1 \ , \\
\dot{c}_2 &= k_1 c_3 - k_3 c_2 c_4 + s_2 \ , \\
\dot{c}_3 &= k_3 c_2 c_4 - k_1 c_3 \ , \\
\dot{c}_4 &= k_2 c_1 - k_3 c_2 c_4 \ .
\end{aligned}$$

The unit for time is seconds and for concentrations the number of molecules per cm^3. Note that an emission source s_2 for NO has been added and that oxygen is taken constant, which is a natural assumption. Typically, the first of the reactions is photochemical, giving rise to rapid changes in concentration values at sunset and sunrise. Figure 1 illustrates this, showing the time evolution of the concentrations for the specific set of initial and source values

$$c = [\,0,\ 1.3 \times 10^8,\ 5.0 \times 10^{11},\ 8.0 \times 10^{11}\,]^T, \quad s_2 = 10^6 \ .$$

The reaction coefficients are defined by

$$k_3 = 10^{-16},\ k_2 = 10^5,\ k_1 = 10^{-40}\ \text{(nighttime)}, \quad k_1 = 10^{-5}\, e^{7.0\, s(t)}\ \text{(daytime)} \ ,$$

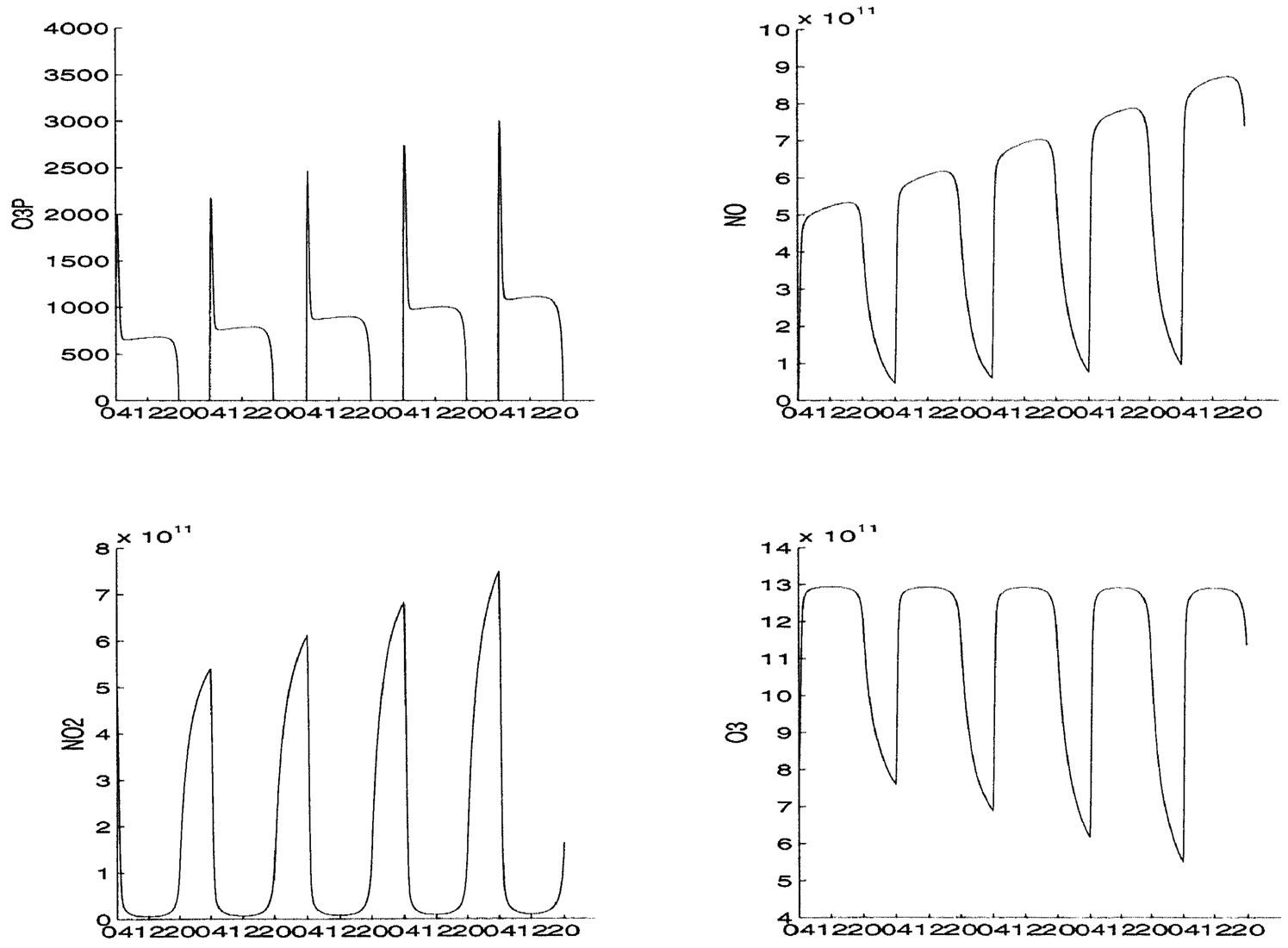


Fig. 1. Time evolution of the concentrations of $O(^3P)$, NO, NO_2, and O_3 for the example problem in Sect. 3.1. The numbers 04, 12, and 20 refer to sunrise, noon, and sunset, respectively

where

$$s(t) = \left(\sin\left(\frac{\pi}{16}(t_h - 4)\right)\right)^{0.2}, \quad t_h = th - 24\lfloor th/24 \rfloor,\ th = t/3600\ ,$$

with daytime between 4 o'clock in the morning and 8 o'clock in the evening and $\lfloor th/24 \rfloor$ denotes the largest integer $\leq th/24$. Hence $s(t)$ is periodic with a period of 24 h, but defined only during daytime.

The concentration values and reaction coefficients more or less approximate their counterparts used in real models. Because oxygen is held constant, k_2 contains the total number of O_2 molecules per cm^3 and is therefore much larger than k_1 and k_3. In all real models such large differences in magnitude between reaction coefficients occur, revealing stiffness. Here the (artificial) time dependence chosen for k_1 is meant to generate the photochemistry. Sunrise and sunset take place at 4 o'clock in the morning and 8 o'clock in the evening. The maximum for k_1 is equal to approximately 0.01 and occurs at 12 o'clock. The concentrations are plotted over approximately 5 days, from sunrise at day 1 (initial value) until sunset at day 5 (time interval $14400 \leq t \leq 517600$ s). Although the model is too limited from the chemical point of view, the variation of the concentrations due to the diurnal cycle are more or less realistically simulated. An important practical point is that the diurnal cycle puts a limit to the step sizes that can be taken in a numerical integration process, say, at most about 30 to 60 min, even for the most sophisticated ODE solver. In real models the number of species differs per application. In their survey paper, Peters et al. [67] pointed out that as many as 40 to 100 species are necessary for an adequate analysis of perturbations to atmospheric chemistry caused by anthropogenic emissions. Realistic box-models containing these numbers of species were used in numerical comparisons reported in [79], [78], [103] and [102].

In all models the chemical-kinetics system can be cast in the production-loss ODE form

$$\dot{c} = R(c), \quad R(c) = P(c) - L(c)c\ , \tag{3}$$

where $P(c)$ contains production terms and $L(c)c$ loss terms with $L(c)$ being a diagonal matrix. Very often R is a quadratic vector function, since mostly only first- and second-order reactions are modelled. All components of $P(c)$ and entries of $L(c)$ are nonnegative, for any nonnegative $c \in \mathbf{R}^m$. From $P(c) \geq 0$, it follows that any solution $c(t) \in \mathbf{R}^m$ is nonnegative:

$$c(t_0) \geq 0 \Rightarrow c(t) \geq 0 \quad \text{for all } t > t_0\ . \tag{4}$$

A numerical solution must maintain this property, since otherwise severe instability can show up, even resulting in a numerical blowup within a few time steps. The simple nonlinear scalar problem

$$\dot{c} = \lambda c^2, \quad \lambda < 0\ , \tag{5}$$

whose solution

$$c(t) = \frac{c(0)}{1 - \lambda c(0) t}$$

remains positive if $c(0)$ is positive, reveals the danger of a negative solution value. Its scalar Jacobian $2\lambda c$ can become large positive if c is negative and $\lambda \ll 0$. In this situation the problem thus can become severely unstable and its solution may not even exist. Henceforth we write positivity when we mean $c \geq 0$.

Associated with R we also have molecular-mass conservation laws

$$\sum_{l=1}^{m} \gamma_l R_l(c) = 0, \quad \forall c \in \mathbf{R}^m \,, \tag{6}$$

with constants $\gamma_l \geq 0$ on the whole set of species c_l (all $\gamma_l > 0$) or on a subset. This trivially implies that for any true solution the mass of the associated set of species is conserved, i.e.,

$$\sum_{l=1}^{m} \gamma_l c_l(t) = \text{constant} \,. \tag{7}$$

It is obviously desirable that a numerical solution should maintain this property.

Curiously, the only solid method known to the authors which mimics both (4) and (7) without any step size restriction, is the well-known implicit Euler method

$$c^{n+1} = c^n + \tau R(c^{n+1}) \,. \tag{8}$$

Here c^n denotes the approximation to $c(t_n)$ and τ is the step size such that $t_{n+1} = t_n + \tau$. The mass conservation property (7) follows immediately for the sequence $\{c^n\}$ by inserting the rule (6) in (8). Positivity can be shown by rewriting (8) as

$$c^{n+1} = \left(I + \tau L(c^{n+1})\right)^{-1} \left(c^n + \tau P(c^{n+1})\right) \,, \tag{9}$$

due to the positivity of P and L. It is here assumed that the backward Euler solution exists and somehow can be solved for any $\tau > 0$. Apparently, the simultaneous preservation of both mass and positivity is cumbersome and often requires fits and tricks if one is not content with the implicit Euler rule or encounters difficulties in using it. Numerical positivity is the most difficult of the two as it is subject to a fundamental constraint. Namely, when applied to a certain class of stiff linear ODE systems, the consistency order of any Runge–Kutta or linear multistep method is bounded by one when such a method is required to be positive for any $\tau > 0$ [15].

Conservation of mass is in principle easy, since any classic explicit or implicit method of Runge–Kutta and multistep type conserves mass. However, implicit methods then need to be properly implemented with regard to solving the implicit relations. For example, modified Newton iterations conserve mass when the analytic Jacobian matrix $R'(c)$ is used. But this property can be lost when iterating with crude approximations to $R'(c)$ or with the alternative iteration techniques exploited in the special-purpose methods discussed in Sect. 3.2.

To illustrate the mass conservation property when iterating with the analytic Jacobian matrix, let us apply the modified Newton method to the implicit Euler scheme (8). Suppressing $n+1$, we first define

$$G(c) = c - c^n - \tau R(c) \ .$$

Modified Newton iteration applied to the implicit Euler system $G(c) = 0$ yields the relation

$$G'(\tilde{c})c_{\text{new}} = G'(\tilde{c})c_{\text{old}} - G(c_{\text{old}}) \ . \tag{10}$$

The solution value $\tilde{c}$ denotes an arbitrary approximation (Newton iteration has $\tilde{c} = c_{\text{old}}$). Suppose that the iterate c_{old} and the Euler solution c^n are mass conservative. Denote $\vec{\gamma} = [\gamma_1, \ldots, \gamma_m]$. Using inner product notation, the mass conservation law (6) then implies $\vec{\gamma} \cdot R(c_{\text{old}}) = \vec{\gamma} \cdot G(c_{\text{old}}) = 0$. For the analytic Jacobian, the mass conservation property implies $\vec{\gamma} \cdot R'(\tilde{c}) = \vec{0}$, where $\vec{0}$ is the zero row vector. Multiplying (10) by $\vec{\gamma}$ now immediately reveals that $\vec{\gamma} \cdot c_{\text{new}} = \vec{\gamma} \cdot c_{\text{old}}$. This proves mass conservation.

In assessing stiff ODE solvers for atmospheric applications, one has to take into account that they will be used within operator splitting (see Sect. 5.2) or variants thereof. This means integration over thousands to millions of grid points, repeatedly in the course of time over the chosen split intervals. If the ODE integration will be carried out with variable step sizes, methods should start up very efficiently since any splitting step involves a start-up. If fixed step sizes will be used, methods should be able to skip artificial stiff transients with a step size large enough to be of practical interest. These artificial transients are introduced by splitting the PDE problem.

Although in the public domain an interesting variety of well-documented stiff ODE solvers is available [35], many atmospheric modellers still use their own variants of the somewhat awkward special-purpose methods of Sect. 3.2. The general experience is that for large-scale three-space-dimensional transport-chemistry problems off-the-shelf solvers are not fast enough and tailor-made solvers are therefore popular. In Sect. 3.3 we will present (sparse) Rosenbrock methods from the stiff ODE field which are not special purpose and certainly competitive.

3.2 Special-purpose methods

The production-loss form (3) is often exploited in the numerical solution process, e.g., in the many variants of the well-known QSSA method, long ago already advocated by Hov and co-workers [37] and Young and Boris [113], in EBI [36] and in Twostep [100]. In this section we will discuss the ideas behind these special-purpose methods. We use the term special purpose in view of the fact that these methods solve the stiff ODE in a sort of explicit manner, in contrast to the requirement of (linearly) implicitness imposed on common stiff ODE methods [21, 35].

3.2.1 Quasi-steady-state approximation

The basic formula for the QSSA (quasi-steady-state approximation) method is

$$c^{n+1} = \mathrm{e}^{-\tau L^{n+1}} c^n + (I - \mathrm{e}^{-\tau L^{n+1}})(L^{n+1})^{-1} P^{n+1} . \tag{11}$$

Here $L^{n+1} = L(c^{n+1})$, etc. QSSA is motivated by the fact that it is exact if P and L are constant, componentwise. In general this is not the case and then the relation is implicit and nonlinear, so iteration is required to find c^{n+1}. Starting from the initial iterate c^n, functional iteration is applied for approximating c^{n+1}. This renders the scheme explicit. Only exponential functions need to be calculated, which in turn could be well approximated by a simple rational Padé approximation. Scheme (11) provides a first-order accurate approximation for $\tau \to 0$. It is inaccurate but often surprisingly stable, owing to the scalar implicitness of the loss term. It is also positive for any $\tau > 0$, something which is an asset for stability. However, (11) does not not preserve mass and problem-dependent mass fixes are really needed to render it competitive. Consequently, QSSA implementations are not very transparent.

As a rule, the number of functional iterations may vary per component. One usually distinguishes between two sets of components and in addition simultaneously switches between different formulas, see, e.g., the description in ref. [103]. To the first set only one iteration is applied. This yields

$$c_l^{n+1} = \mathrm{e}^{-\tau L_l^n} c_l^n + \left(I - e^{-\tau L_l^n}\right)\left(L_l^n\right)^{-1} P_l^n . \tag{12}$$

If for a component in this first set $\tau L_l^n < 0.01$, say, then for this component $\mathrm{e}^{-\tau L_l^n}$ is replaced by $1 - \tau L_l^n$, resulting in the explicit Euler formula

$$c_l^{n+1} = c_l^n + \tau\left(P_l(c^n) - L_l(c^n)c_l^n\right) .$$

If for a component in this set $\tau L_l^n > 10.0$, say, then $\mathrm{e}^{-\tau L_l^n}$ is replaced by zero, resulting in the steady-state solution expression

$$c_l^{n+1} = P_l^n / L_l^n .$$

To the second set of components more such iterations are applied, say, about 5. The selection of these two different sets is based on chemical considerations. Components for which a strong nonlinear coupling exists belong to the second set. Note that the replacements of the exponential function are not essential, since the exponential function automatically takes care of these replacements with sufficiently large accuracy.

Of considerable greater importance is that in order to be efficient, the scheme must be heavily tuned by "lumping". This technique imposes conservation for subsets of species, thus aiming at eliminating strong couplings among components, since the scalarly implicit approach is more suitable for problems with a weak coupling. To this end, subsets are grouped together into a new species which can be solved accurately. For example, for model (2) the species $\mathrm{NO}_x = \mathrm{NO} + \mathrm{NO}_2$ satisfies the

differential equation $\mathrm{d}/\mathrm{d}t\,[\mathrm{NO}_x] = s_2$, so that NO_x satisfies the conservation law

$$[\mathrm{NO}_x](t) = [\mathrm{NO}_x](t_0) + s_2(t - t_0)\ .$$

This law thus can be imposed after each integration step by either redefining [NO] or $[\mathrm{NO}_2]$. With regard to this choice, care should be exercised in avoiding cancellation of digits by subtracting nearly equal, very large concentration values. Hence it may depend on the time of the day which species is redefined. In a similar vein one may choose to truly eliminate NO or NO_2 from the integration.

In practice the lumping technique is intertwined with the functional iteration procedure. Needless to say that for large models lumping requires considerable chemical knowledge of how to group species. Normally, for a given problem the technique can only be described comprehensively in chemical notation. We believe that the increasing complexity of models will impede tuning. It should be emphasized though that atmospheric chemists have developed very efficient, tuned implementations which for low accuracies are really fast. The low accuracy means no restriction for the field. An example is provided by a QSSA implementation for the so-called EMEP chemistry consisting of ca. 140 reactions among ca. 70 species, see [103] and the references therein to Hov and coworkers. We note that the CHEMEQ scheme proposed in ref. [113] is similar to the QSSA scheme discussed here. Attempts to improve QSSA by numerical considerations have been reported in refs. [45] and [101]. The resulting schemes provide only a minor improvement and are not advocated without the aforementioned lumping. In Sect. 3.4 results of a numerical comparison will be presented.

3.2.2 EBI

The EBI (Euler backward iterative) method proposed by Hertel et al. [36] is based on the implicit Euler rules (8) and (9). It improves the QSSA approach in two ways. First, the positive implicit Euler method preserves mass and, unlike the exponential formula (11), implicit Euler has the correct local error structure for stiff problems (see refs. [21] and [101] for details). Second, the EBI iteration procedure is better than the simple functional iteration procedure used by QSSA. The EBI procedure is a mix of block Jacobi techniques. Within the blocks strongly coupled species are kept together in the iteration procedure, even using exact solutions for subsystems within the blocks. However, like QSSA the method is heavily problem dependent. A complete specification is possible only in chemical notation. It is also noted that the EBI iterates themselves are not mass conservative. Hence with only a few iterates mass may get lost or produced in the course of time. Yet, for the chemistry schemes treated, the EBI scheme has been shown to be very fast for low accuracies. We recall that low accuracy means no restriction for the field. Accuracy can also be improved by porting the specific iteration procedure to a higher-order integration formula, e.g., to the second-order backward differentiation formula (BDF2) used by Twostep (see below). In Sect. 3.4 the result of a numerical comparison will be presented.

3.2.3 Twostep

The variable-step-size solver Twostep was proposed by Verwer [100] and further elaborated in collaboration with Simpson [103]. Like the QSSA method, it is motivated by the production-loss form (3). The main feature of Twostep is that a simple form of nonlinear Gauss–Seidel iteration is applied to the implicit BDF2 method, rather than the usual modified Newton iteration. Consider the 2-step BDF

$$c^{n+1} = \tfrac{4}{3}c^n - \tfrac{1}{3}c^{n-1} + \tfrac{2}{3}\tau R(c^{n+1}) , \tag{13}$$

for simplicity in fixed-step-size form. Applied to the production-loss form (3) we can write

$$\begin{aligned} c^{n+1} = G(c^{n+1}) &:= \left(I + \tfrac{2}{3}\tau L(c^{n+1})\right)^{-1}\left(C^n + \tfrac{2}{3}\tau P(c^{n+1})\right) , \\ C^n &= \tfrac{4}{3}c^n - \tfrac{1}{3}c^{n-1} . \end{aligned} \tag{14}$$

The Gauss–Seidel technique is now applied to the nonlinear system of equations $c = G(c)$ as follows. Given the vector iterate $c^{(i)}$, we apply the componentswise formula

$$c_l^{(i+1)} = G_l\left(c_1^{(i+1)}, \ldots, c_{l-1}^{(i+1)}, c_l^{(i)}, \ldots, c_m^{(i)}\right), \quad l = 1, \ldots, m , \tag{15}$$

defining an explicit Gauss–Seidel process which requires only scalar divisions. The choice for BDF2 is justified in view of the low-accuracy requirement. The approach can of course be applied also for higher-order BDF formulas, or to the implicit Euler formula. In computational effort per time step, Twostep is more or less comparable to QSSA and EBI.

For components for which both P_l and L_l are constant, the implicit solution is obtained in one iteration. Consequently, when individual components rapidly approach their steady-state value P_l/L_l, they are handled efficiently. In this respect we have a resemblance with QSSA. The difference is that we start from a mass conservative formula, which does not a priori assume that P_l and L_l slowly vary, in contrast with QSSA. This provides a better starting point for components for which P_l and L_l are truly varying, supposing that the Gauss–Seidel technique then still yields a sufficiently good approximation within a few iterations.

The technique appears to be very stable for gas-phase chemistry problems, rendering it feasible for low accuracies. In this respect we have not observed much difference with the more commonly used iterative modified Newton technique. However, the latter preserves mass when based on the analytical Jacobian matrix $R'(c)$, which is not true for Gauss–Seidel. So the BDF2 solution should be solved sufficiently accurate to preserve mass sufficiently accurate. It should also be noted that BDF2 itself does not guarantee positivity. Comprehensive comparisons reported in refs. [78], [79] and [102] show the effectiveness of Twostep compared to QSSA, EBI, and a number of general-purpose solvers from the stiff ODE field, including LSODE, VODE, RODAS, and SEULEX (see ref. [35] for descriptions). In these comparisons Twostep outperforms QSSA (both without tuning) and is often faster than

the general-purpose solvers in the low-accuracy region. Unfortunately, only one comparison with EBI exists [79]. In this comparison EBI and Twostep are more or less competitive to one another. In Sect. 3.4 some results of these comparisons will be presented. Finally it is stressed that Twostep is meant for gas-phase problems only. For aqueous-phase, heterogeneous chemistry exhibiting a much stronger coupling between components, the Gauss–Seidel convergence is too slow for step sizes of practical interest. This also holds for QSSA. We are not aware of an EBI implementation for heterogeneous chemistry problems.

Iteration (15) is nonlinear and works matrix-free, i.e., it does not need the Jacobian matrix. A related linear Gauss–Seidel technique, which does use the Jacobian matrix, has been proposed [51]. The two techniques are believed to be close in performance. See ref. [51] for numerical comparisons.

3.3 General-purpose methods of Rosenbrock type

We next discuss an alternative for the special-purpose methods, viz., low-order Rosenbrock methods from the stiff ODE field which use sparse matrix routines to reduce numerical algebra overhead [77–79]. Earlier experiences [35] have shown that for stiff ODEs from different applications Rosenbrock solvers are competitive with other solvers for low to modest accuracies. Because for atmospheric applications the greatest interest lies in high efficiency for very low accuracy (two digits at most), low-order Rosenbrock methods are natural candidates. The use of a fourth-order Rosenbrock method to simulate pollutant transport and flow in rivers is discussed in ref. [71].

3.3.1 Rosenbrock method

Usually, stiff solvers use some form of implicitness in the discretization formula for reasons of numerical stability. The simplest implicit scheme is the implicit Euler method (8). Since c^{n+1} is defined implicitly, this numerical solution itself must also be approximated. Common practice is to use some modification of the iterative Newton method, again for reasons of numerical stability. Suppose, for the time being, that the problem is autonomous, i.e., R is independent of time t. Next suppose that just one iteration per time step is applied. If we then assume that c^n is used as the initial iterate, the following numerical result is found:

$$c^{n+1} = c^n + k, \quad \text{where } k = \tau R(c^n) + \tau A k \tag{16}$$

and A denotes the Jacobian matrix $R'(c^n)$. The numerical solution is now effectively computed by solving the system of linear algebraic equations that defines the increment vector k, rather than a system of nonlinear equations.

Rosenbrock [75] proposed to generalize this linearly implicit approach to methods using more stages, so as to achieve a higher order of consistency. The crucial

consideration put forth was to no longer use the iterative Newton method, but instead to derive stable formulas by working the Jacobian matrix directly into the integration formula. His idea has found widespread use, and a generally accepted formula for a so-called s-stage Rosenbrock method is

$$c^{n+1} = c^n + \sum_{i=1}^{s} b_i k_i, \quad \text{where } k_i = \tau R(c^n + \sum_{j=1}^{i-1} \alpha_{ij} k_j) + \tau A \sum_{j=1}^{i} \gamma_{ij} k_j \qquad (17)$$

and s and the formula coefficients b_i, α_{ij}, and γ_{ij} are chosen to obtain a desired order of consistency and stability for stiff problems. A brief introduction on the properties of consistency, stability, and stiff accuracy for Rosenbrock methods is presented in the appendix of ref. [79]. A comprehensive treatment of Rosenbrock methods can be found in Hairer and Wanner [35].

For the nonautonomous system $\dot{c} = R(t, c)$, the definition of k_i is changed to

$$k_i = \tau R(t_n + \alpha_i \tau, c^n + \sum_{j=1}^{i-1} \alpha_{ij} k_j) + \gamma_i \tau^2 \frac{\partial R}{\partial t}(t_n, c^n) + \tau A \sum_{j=1}^{i} \gamma_{ij} k_j \ ,$$

where $\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}$ and $\gamma_i = \sum_{j=1}^{i} \gamma_{ij}$. This nonautonomous formulation follows from the autonomous formula (17) by treating time t also as a dependent variable, i.e., by augmenting the system with the equation $\dot{t} = 1$. Like Runge–Kutta methods, Rosenbrock methods successively form intermediate results

$$c^n + \sum_{j=1}^{i-1} \alpha_{ij} k_j, \quad 1 \le i \le s \ , \qquad (18)$$

which approximate the solution at the intermediate time points $t_n + \alpha_i h$. Rosenbrock methods are therefore also called Runge–Kutta–Rosenbrock methods. Observe that if we identify A with the zero matrix and omit the $\partial R/\partial t$ term, a classical explicit Runge–Kutta method results. The coefficients γ_{ii} are often taken equal for all stages, i.e., $\gamma_{ii} = \gamma$ for all $i = 1, \ldots, s$, because we then encounter only one matrix $I - \gamma \tau A$. This saves computational work (for LU decompositions). Note for $s = 1$ and $\gamma = 1$ the above linearized implicit Euler formula is recovered.

Rosenbrock methods are attractive for a number of reasons. Like fully implicit methods, they mimic exact conservation properties due to the use of the analytic Jacobian matrix. However, they do not require an iteration procedure as for truly implicit methods and are therefore more easy to implement. So they are of nonit-erative type, which is an asset for parallel implementations over space grids as true iteration may create load imbalancing. Rosenbrock methods can be developed to possess optimal linear stability properties for stiff problems (A- and L-stability). They are of one-step type and thus can rapidly change step size. We recall that this is of particular importance for our application in view of the many operator-split restarts when variable step sizes are to be used. But perhaps of greater importance

is that a suitable Rosenbrock method can be applied to a much wider range of chemical-kinetics problems than special-purpose methods. Test results in refs. [79] and [106] indicate that the methods RODAS3 and ROS2, discussed below, can be applied equally efficiently to tropospheric-stratospheric gas-phase and aqueous, heterogeneous problems. In this regard a suitable Rosenbrock method can be called a general-purpose method.

Each time step requires an evaluation of A, s matrix-vector multiplications with A, and assuming that $\gamma_{ii} = \gamma$, s solutions of a linear system with the matrix $I - \gamma\tau A$ accompanied with s derivative evaluations. The multiplications with A are easily avoided (see [35: sect. IV.7]). Most of the CPU time is spent on the solution of the s linear systems involving the matrix $I - \gamma\tau A$. The linear system solution requires one matrix factorization (LU decomposition) and s backsolves (forward-backward substitutions).

High CPU costs for these matrix manipulations are of course most notable if the number of species m is large. Fortunately, for large atmospheric chemistry models the number of zeroes in A readily exceeds 50% by far, since in large models species react with only a limited number of others. This high level of sparsity can be exploited to significantly reduce the costs for the Jacobian update, LU decomposition, and backsolves. For atmospheric problems, sparsity has been considered in refs. [44], [77–79], [102], [106], and [96]. In particular, a pivoting technique using a diagonal Markowitz criterion proves to minimize fill-in and to reduce the numerical algebra costs considerably for large models [77].

To relieve modellers from a lot of error-prone coding, for a long time symbolic preprocessing is used to automatically turn a set of chemical reactions into code defining the ODE system. More sophisticated preprocessors also generate code (Fortran mostly) for the Jacobian, the LU decomposition and the backsolve. Such a preprocessor is KPP (kinetics preprocessor) [20], which outputs sparse matrix routines based on the diagonal Markowitz technique advocated in ref. [77]. The routine for the backsolve comes without indirect addressing. The LU routine delivered by the original KPP uses indirect addressing. This can be easily avoided [88], which is computationally advantageous. Altogether this means that with KPP the numerical algebra within a Rosenbrock method can be handled very efficiently.

Below we present a method of order two and a method of order three, which both have proven to be efficiently applicable to atmospheric chemical-kinetics problems. For low accuracies, both are truly competitive to BDF solvers like LSODE, VODE, and SPRINT [35]. While their costs per time step can still be significantly higher than for the explicit special-purpose methods of Sect. 3.2, as a rule they outperform these special-purpose methods.

3.3.2 ROS2

As mentioned previously, a major concern is positivity. Truly positive Rosenbrock methods for general nonlinear chemical-kinetics systems (3) do not exist. Yet, in a

search for low-order Rosenbrock methods with acceptable positivity and nonlinear stability properties, the following second-order consistent method proved promising [106]:

$$
\begin{aligned}
c^{n+1} &= c^n + \tfrac{1}{2}k_1 + \tfrac{1}{2}k_2 \,, \qquad (19)\\
k_1 &= \tau R(c^n) + \gamma\tau A k_1 \,,\\
k_2 &= \tau R(c^n + k_1) - 2\gamma\tau A k_1 + \gamma\tau A k_2 \,.
\end{aligned}
$$

Henceforth this method will be called ROS2. With $\gamma = 1 + 1/\sqrt{2}$ it is L-stable, which is of obvious interest for stiff chemistry. The property of mass conservation holds as long as the true analytic Jacobian matrix R' is used. Curiously, compared to the third-order method RODAS3 given below, ROS2 appears to be much more stable for nonlinear problems for large fixed step sizes. See ref. [106] for a stability test with three different chemistry models from practice. A rigorous explanation is lacking. We only know that ROS2 is positive for the scalar models $\dot{c} = \lambda c$ and $\dot{c} = \lambda c^2$ for any $\tau > 0$ and any $\lambda < 0$, whereas this does not hold for RODAS3.

ROS2 also has another advantage compared to RODAS3, viz., it is second-order consistent for any matrix A. This "approximate Jacobian" or "W-method" property [35], can be used to advantage in the solution of 3-D transport-chemistry problems. This will be explained in Sect. 5.4. In Sect. 3.4 some illustrative results of numerical comparisons between ROS2 and other methods will be presented. Finally, ROS2 is currently examined for implementation in the regional tropospheric models EURAD [48] and LOTOS [14].

3.3.3 RODAS3

In a comparison conducted by Sandu et al. [79] the third-order Rosenbrock method RODAS3 belonged to the best methods tested there. RODAS3 was designed along the same principles as RODAS from ref. [35], the only difference being its lower order and lower number of stages. RODAS3 is based on a stiffly accurate, embedded pair of consistency order 3 (2). The number of stages is $s = 4$, requiring four backsolves, but only three derivative evaluations are used. The third-order formula is given by

$$
\begin{aligned}
c^{n+1} &= c^n + \tfrac{5}{6}k_1 - \tfrac{1}{6}k_2 - \tfrac{1}{6}k_3 + \tfrac{1}{2}k_4 \,, \qquad (20)\\
k_1 &= \tau R(c^n) + \tfrac{1}{2}\tau A k_1 \,,\\
k_2 &= \tau R(c^n) + \tau A k_1 + \tfrac{1}{2}\tau A k_2 \,,\\
k_3 &= \tau R(c^n + k_1) - \tfrac{1}{4}\tau A k_1 - \tfrac{1}{4}\tau A k_2 + \tfrac{1}{2}\tau A k_3 \,,\\
k_4 &= \tau R(c^n + \tfrac{3}{4}k_1 - \tfrac{1}{4}k_2 + \tfrac{1}{2}k_3) + \tfrac{1}{12}\tau A k_1 + \tfrac{1}{12}\tau A k_2 - \tfrac{2}{3}\tau A k_3 + \tfrac{1}{2}\tau A k_4 \,.
\end{aligned}
$$

Like ROS2, this integration formula is L-stable and preserves mass, since the true analytic Jacobian matrix is supposed. Unfortunately, RODAS3 is not positive. The

scalar stability test model $\dot{c} = \lambda c$, yielding the numerical solution

$$R(z) = \frac{1 - \frac{1}{2}z - \frac{1}{4}z^2}{(1 - \frac{1}{2}z)^3}, \quad z = \tau\lambda \ ,$$

provides a simple counterexample for $z < -1 - \sqrt{5}$. When using variable step sizes and local error control, as in ref. [79], serious lack of positivity will automatically lead to step rejections and smaller step sizes then will save the computation, provided the control is robust enough. If this occurs not too often, the damage is limited. When fixed step sizes are used, lack of positivity can lead to a numerical blowup. Some numerical results for RODAS3 will be presented in Sect. 3.4.

3.4 Numerical illustration

We throw a quick glance at performances of QSSA, EBI, Twostep, RODAS3, and three other solvers, as shown in two work-precision diagrams from ref. [79]. See Figs. 2 and 3, which are identical to figs. 1 and 3 in ref. [79]. The captions are unchanged and mention names of chemistry box-models of tropospheric gas-phase type and different solvers. For solvers not discussed here, see refs. [79] and [35]. (Results for ROS2 are not included either since this method was examined after the publication of ref. [79]. The report [14] is completely devoted to ROS2, with and without use of "approximate Jacobians" as proposed in ref. [106].) QSSA only occurs in Fig. 3. It is the most simple QSSA method tested in ref. [78], without any tuning. EBI only occurs in Fig. 2.

The solvers are applied in an operator splitting setting with a large number of restarts. The chemistry model TMk from Fig. 2 is limited in size, consisting of 36 reactions between 18 species. In Fig. 3 two chemistry models appear, viz., CBM-IV (81 reactions and 32 species) and AL (176 reactions and 84 species). So the latter one is quite large. In Figs. 2 and 3, computational work is measured in CPU times along the horizontal axis in a logarithmic scale. Precision is measured along the vertical axis, also in a logarithmic scale through the number of correct digits, here called SDA. A relative error norm is used [79]. SDA $= k$ means $10^{-k} \cdot 100\%$ accuracy in this norm. In the actual practice, efficient codes should roughly yield between 1% (SDA $= 2$) and 10% (SDA $= 1$) accuracy for a minimum in CPU. It is stressed that SDA > 2 is not of much interest for practice, while SDA < 1 is too inaccurate.

When looking at Figs. 2 and 3, it is interesting to note that we show results for two sorts of solvers, standard solvers from the stiff ODE field provided with efficient sparse matrix routines and special-purpose solvers specifically designed for chemistry computations in big operational atmospheric-transport codes using operator splitting. The standard solvers are the Rosenbrock solvers RODAS3 [79], ROS3 [79], and RODAS [35], the (Gear type) backward differentiation solver VODE, and the extrapolation solver SEULEX (see [35] and the references therein to VODE

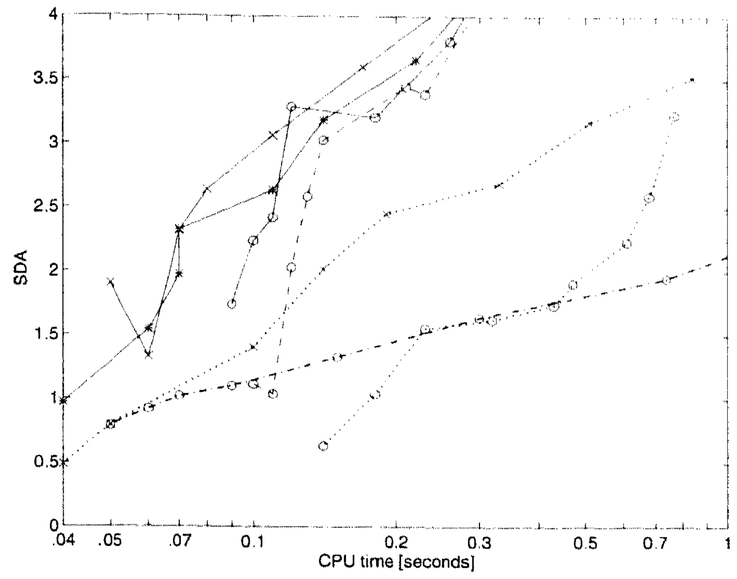
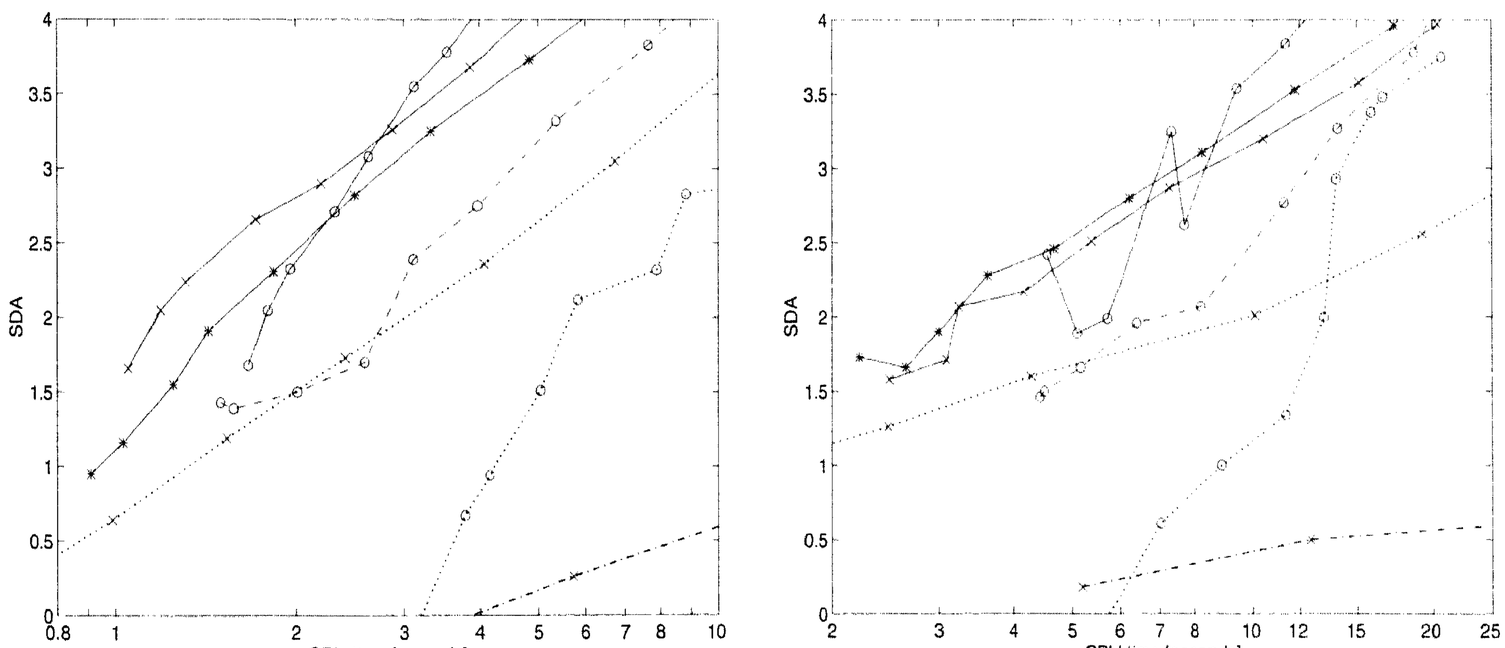


Fig. 2. Work-precision diagram for solvers applied to test problem TMk. $-*-$, sparse RODAS3; $-\times-$, sparse ROS3; $-\circ-$, sparse RODAS; $\cdot\cdot\times\cdot\cdot$, Twostep Seidel; $\cdot\cdot\circ\cdot\cdot$, sparse VODE; - - ○ - -, sparse SEULEX; - · - ○ - · - -, EBI

and SEULEX). The special-purpose solvers are QSSA, EBI, and Twostep and were discussed above.

The main conclusion we can draw from Figs. 2 and 3 is that in the low-accuracy range $1 < \mathrm{SDA} < 2$, the (sparse) Rosenbrock solvers RODAS3 and ROS3 are most efficient. The other three (sparse) standard solvers are not competitive in this range. The special-purpose solvers EBI and Twostep also perform quite satisfactorily in the low-accuracy range. Twostep is a reliable solver, even for very crude tolerances, whereas forcing RODAS3 or ROS3 to compute in lower accuracy is sometimes problematical (a too loose local error control can result in instability). The special-



a **b**

Fig. 3. Work-precision diagram for test problems CBM-IV urban (**a**) and AL urban (**b**), with a restart every 15 min. $-*-$, sparse RODAS3; $-\times-$, sparse ROS3; $-\circ-$, sparse RODAS; $\cdot\cdot\times\cdot\cdot$, Twostep Seidel; $\cdot\cdot\circ\cdot\cdot$, sparse VODE; - - ○ - -, sparse SEULEX; - · - ○ - · - -, QSSA

purpose solver QSSA falls behind severely and cannot compete, but it is stressed that lumping is not used here. It should also be stressed that the special-purpose solvers Twostep, EBI, and QSSA have been designed for gas-phase problems and that their performance can drop dramatically when they are applied to aqueous-phase, heterogeneous problems. In this regard standard stiff ODE solvers are much more robust as they solve these more difficult problems with more or less the same efficiency [78].

4 Advection problem

The major transport mechanism in air pollution models is advective transport in a wind field. Hence the quality of the advection computation is crucial. In this section we consider several schemes based on third-order upwind type discretizations with flux limiting, as pioneered by van Leer and others. This choice provides a good compromise between accuracy and shape preservation on the one hand and computational ease on the other hand. For other schemes, such as spectral or finite-element schemes, we refer to refs. [74, 107].

4.1 Semidiscrete and fully discrete advection schemes

For simplicity of presentation we start with the scalar advection problem in one space dimension,

$$c_t + uc_x = 0, \quad u = \text{constant} > 0\,, \tag{21}$$

for $t \geq 0$ with given initial condition. For the moment, boundary conditions will be ignored. The grid is assumed to be uniform with a grid size h, i.e., $x_i = ih$.

4.1.1 Semidiscrete advection scheme

Applying the third-order upwind-biased spatial discretization to c_x gives the semidiscrete system

$$\frac{\mathrm{d}}{\mathrm{d}t}w_i(t) = \frac{u}{h}\left(-\frac{1}{6}w_{i-2}(t) + w_{i-1}(t) - \frac{1}{2}w_i(t) - \frac{1}{3}w_{i+1}(t)\right)\,. \tag{22}$$

Here $w_i(t)$ can be viewed as an approximation to the point value $c(x_i, t)$ or to the average over the cell $(x_{i-1/2}, x_{i+1/2})$. The above spatial discretization scheme is also called the $\kappa = 1/3$-scheme [56]. This system of ODEs requires numerical time integration, for instance by a Runge–Kutta method, to obtain fully discrete approximations. As mentioned in Sect. 1, this approach, in which space and time are considered separately, is called the method of lines (MOL). At this initial stage of derivation, we will first motivate the third-order upwind-biased (upwind for short in the remainder) formula, by relating it with the standard fourth-order central one, and by looking at the modified equation and dissipation and dispersion properties. The time integration will be discussed later on.

Inspection of (22) reveals that it fits in the form

$$\frac{\mathrm{d}}{\mathrm{d}t}w_i(t) + u\frac{w_{i-2}(t) - 8w_{i-1}(t) + 8w_{i+1}(t) - w_{i+2}(t)}{12h}$$
$$= -\gamma u\frac{1}{12}h^3\frac{w_{i-2}(t) - 4w_{i-1}(t) + 6w_i(t) - 4w_{i+1}(t) + w_{i+2}(t)}{h^4},$$

where $\gamma = 1$. For $\gamma = 0$ one gets the fourth-order central scheme. The difference formula in the right-hand side is the standard, central approximation to the fourth-order derivative,

$$\frac{w_{i-2} - 4w_{i-1} + 6w_i - 4w_{i+1} + w_{i+2}}{h^4} = \frac{\partial^4 w}{\partial x^4}(x_i) + \frac{1}{60}h^2\frac{\partial^6 w}{\partial x^6}(x_i) + O(h^4)\,. \tag{23}$$

Hence the upwind scheme is identical to the central one if the two central schemes are applied to

$$c_t + uc_x = -\frac{1}{12}h^3 u\frac{\partial^4 c}{\partial x^4}\,, \tag{24}$$

showing the intimate relation between the two.

This equation is often called a modified equation for the original advection equation (21). The meaning of a modified equation is that in the limit the discrete solution is expected to be closer to the true solution of a modified equation than to that of the original equation to which the discretization is applied. Hence studying modified equations can enhance insight in discretization properties, see, for example, [33]. In our case we see that the term $-c_{xxxx}$ is a higher-order dissipation term, so that the upwind scheme yields artificial damping. However, oscillations are still possible. Note that the equation $c_t = -c_{xxxx}$ does not satisfy the maximum principle. For instance, if $c(x, 0) = 1 - \cos(2\pi x)$, then $c(0, 0) = 0$ and $c_t(0, 0) = -(2\pi)^4 < 0$.

Other insight can be obtained with Fourier analysis. This analysis enables one to study the important properties of numerical dispersion (different frequencies result in different wave speeds) and numerical dissipation (artificial diffusion). Fourier analysis is possible for constant coefficient problems satisfying periodic boundary conditions. The use of Fourier analysis for difference schemes is originally due to J. von Neumann and is well explained in the classic monograph by Richtmyer and Morton [72] and in later textbooks, e.g., [34].

Let the Fourier mode

$$c(x, t) = \hat{c}(t)\,\mathrm{e}^{\sigma\omega x}, \quad \sigma = \sqrt{-1},\ \omega \in \mathbf{R}\,, \tag{25}$$

be a solution of (21). Then $\hat{c}$ solves $\mathrm{d}\hat{c}/\mathrm{d}t + \sigma\omega u\hat{c} = 0$, so that

$$c(x, t) = \hat{c}(0)\,\mathrm{e}^{-\sigma\omega u t}\,\mathrm{e}^{\sigma\omega x}\,, \tag{26}$$

and $\hat{c}(0)$ can be interpreted as a Fourier coefficient arising in a Fourier expansion or decomposition of the initial function $c(x, 0)$, associated with a frequency ω. For linear difference schemes the Ansatz of Fourier–von Neumann analysis is that a

completely similar decomposition in Fourier modes exists and hence that it suffices to study properties of the difference scheme for single Fourier modes, similar to (26).

Let

$$w_i(t) = \hat{w}(t)\,\mathrm{e}^{\sigma\omega x_i}, \quad \sigma = \sqrt{-1}\,,$$

be such a Fourier mode for (22). An elementary calculation shows that $\hat{w}$ solves $\mathrm{d}\hat{w}/\mathrm{d}t = \lambda\hat{w}$, with λ being an eigenvalue of the linear upwind operator, i.e.,

$$\lambda(\xi) = -\frac{1}{3}\frac{u}{h}\big((\cos\xi - 1)^2 + \sigma\,\sin\xi(4 - \cos\xi)\big), \quad \xi = \omega h\,. \tag{27}$$

Hence

$$w_i(t) = \hat{w}(0)\,\mathrm{e}^{\lambda t}\,\mathrm{e}^{\sigma\omega x_i}\,, \tag{28}$$

and it should be noted that we may restrict ξ to the interval $[-\pi, \pi]$, reflecting the fact that only a finite number of frequencies can be represented on the grid.

It is evident that the eigenvalue λ is the semidiscrete counterpart of the purely imaginary eigenvalue $-\sigma\omega u$. The eigenvalue λ determines the dissipation and dispersion properties of the upwind scheme. Dissipation is introduced through the negative real part of λ, which satisfies

$$-\frac{1}{3}\frac{u}{h}(\cos\xi - 1)^2 = -\frac{1}{12}u\omega\xi^3 + O(\xi^5)\,.$$

Hence for given ω it decreases with h^3 for $h \to 0$, in line with what the modified equation shows. Clearly, very-high-frequency modes (ω far away from zero) are rigorously damped. The numerical dispersion is determined by the purely imaginary part of λ, which approximates $-\omega u$ and satisfies

$$\frac{1}{3}\frac{u}{h}\sin\xi(4 - \cos\xi) = (1 + O(\xi^4))\omega u\,.$$

Hence the third-order upwind scheme is of order 4 with respect to dispersion errors, which we may consider to be an asset. In fact, it has exactly the same numerical phase velocity as the fourth-order central scheme, once more revealing the intimate relation between the two (for fourth-order central λ has the same imaginary part and the real part is zero).

Dispersion errors show up in oscillations and hence in negative concentrations. Dispersion errors are unavoidable, no finite-difference method can resolve arbitrarily short wavelengths without excessive dispersion errors. Inspection of λ shows that in the frequency region where the dispersion error is largest also the spurious damping is largest. So also in this respect the third-order upwind scheme is a good choice. But to truly avoid negative concentrations in regions of strong variation, a technique like flux limiting is needed. We will introduce this technique later on in Sect. 4.2. First we will introduce another upwind discretization for (21).

4.1.2 Direct space-time advection scheme

This discretization falls outside the MOL approach as it is based on a direct space-time (DST) discretization. Using the same stencil, involving two upwind points and one downwind point, the discretization formula reads

$$w_i^{n+1} = \gamma_{-2} w_{i-2}^n + \gamma_{-1} w_{i-1}^n + \gamma_0 w_i^n + \gamma_1 w_{i+1}^n \,. \tag{29}$$

Here w_i^n is the approximation to $c(x_i, t_n)$ and $t_{n+1} = t_n + \tau$, with τ being the step size in time. The coefficients, depending on the Courant number

$$\nu = \frac{u\tau}{h} \,, \tag{30}$$

are given by

$$\gamma_{-2} = -\tfrac{1}{6}\nu(1-\nu^2), \quad \gamma_{-1} = \tfrac{1}{2}\nu(2-\nu)(1+\nu), \quad \gamma_0 = \tfrac{1}{2}(2-\nu)(1-\nu^2) \,,$$
$$\gamma_1 = -\tfrac{1}{6}\nu(2-\nu)(1-\nu) \,.$$

This formula also has order 3 and it is stable for $\nu \leq 1$, which follows from the classical, fundamental results on optimal-order schemes of Strang [93] on the basis of Fourier–von Neumann analysis. The method can be derived by the consistency conditions, but also from a semicharacteristic approach, where the characteristics are traced backward in time, from t_{n+1} to t_n, and Lagrange interpolation is used. From the semicharacteristic interpretation it follows that the method becomes more accurate the closer ν is to 1. Specifically, the global error $|w(x_i, t_n) - w_i^n|$ can be bounded by $C(1-\nu)h^3$, with C being a positive constant depending on the exact solution only, see [40].

There is a close connection between (22) and (29). If we consider the limit $\tau \to 0$ for fixed h and $t_n > 0$, then the approximations w_i^n of (29) converge to the solution $w_i(t_n)$ of (22). This easily follows by writing (29) as

$$\frac{1}{\tau}(w_i^{n+1} - w_i^n) = \frac{u}{h}\left(-\tfrac{1}{6}w_{i-2}^n + w_{i-1}^n - \tfrac{1}{2}w_i^n - \tfrac{1}{3}w_{i+1}^n\right) + O(\tau), \quad h > 0, \ \tau \to 0 \,.$$

So, for h fixed and ν tending to zero, the DST scheme becomes identical to the MOL scheme with exact time integration. Recall, however, that there is a clear advantage for the DST scheme since its error will become smaller for larger ν, up to $\nu = 1$. In the MOL approach the error will grow with increasing ν due to growing temporal inaccuracy of the numerical ODE method. The advantage can also be illustrated by directly expanding (29) (see, for instance, [40]). This gives

$$c_t + uc_x = -\frac{1}{12}e(\nu)h^3 u\frac{\partial^4 c}{\partial x^4} + O(h^4) \,, \tag{31}$$

where $e(\nu) = (1-\nu)(1-\frac{1}{2}\nu)(1+\nu)$. Apparently, $e(\nu)$ vanishes as $\nu \to 1$. Observe that in line with the convergence of (29) to (22) shown above, the modified equation (24) is recovered if we replace $e(\nu)$ by $e(0) = 1$ and remove the $O(h^4)$ term.

4.2 Flux limiting

As already mentioned, both methods (22) and (29) can produce some over- and undershoot, which may lead to negative values. If w stands for a concentration of a chemical species, this should, of course, remain nonnegative. Although the negative values created by the upwind schemes are rather small compared to those created by central difference schemes, they can lead to instabilities when advection is combined with chemical reactions. Schemes with the property that solutions remain nonnegative for arbitrary nonnegative initial values are often called positive. The simplest way to avoid negative values, setting them equal to zero, can lead to a distorted mass balance. Following the approach of Sweby [97], we therefore consider flux limiting.

We start with the DST scheme, which can be written in conservation form as

$$w_i^{n+1} = w_i^n + \frac{\tau}{h}(f_{i-1/2}^n - f_{i+1/2}^n) \,. \tag{32}$$

Omitting the superscripts n on the right-hand side, the fluxes $f_{i+1/2}$ are given by

$$f_{i+1/2} = u\big(-\tfrac{1}{6}(1-\nu^2)w_{i-1} + \tfrac{1}{6}(1+\nu)(5-2\nu)w_i + \tfrac{1}{6}(2-\nu)(1-\nu)w_{i+1}\big) \,.$$

Let $d_0 = \frac{1}{6}(2-\nu)(1-\nu)$ and $d_1 = \frac{1}{6}(1-\nu^2)$. Introducing the ratios

$$\theta_i = \frac{w_i - w_{i-1}}{w_{i+1} - w_i} \,,$$

the flux can be written as

$$f_{i+1/2} = u\big(w_i + (d_0 + d_1\theta_i)(w_{i+1} - w_i)\big) \,. \tag{33}$$

For limiting we consider the more general form

$$f_{i+1/2} = u\big(w_i + \psi(\theta_i)(w_{i+1} - w_i)\big) \,, \tag{34}$$

with a so-called limiter function ψ yet to be determined.

Since $w_{i+1} - w_i = (w_i - w_{i-1})/\theta_i$, the total scheme can be written in the form

$$w_i^{n+1} = (1 - \nu\phi_i)w_i + \nu\phi_i w_{i-1} \quad \text{with } \phi_i = 1 - \psi(\theta_{i-1}) + \frac{1}{\theta_i}\psi(\theta_i) \,, \tag{35}$$

from which we immediately conclude that the method is positive if and only if

$$0 \le \nu\phi_i \le 1 \,.$$

The limiter function ψ should now be chosen such that this requirement holds for all ratio values θ_i. On the other hand, the original nonlimited flux should be reobtained for ratios θ_i close to 1, since this is the generic situation for a smooth profile. As in refs. [40] and [41] we consider

$$\psi(\theta) = \max(0,\ \min(1, d_0 + d_1\theta, \mu\theta)) \,, \tag{36}$$

where μ is a positive parameter, which is still free. With this limiter we have $0 \le$

$\psi(\theta) \leq 1$ and $0 \leq \psi(\theta)/\theta \leq \mu$. Hence $0 \leq \phi_i \leq 1 + \mu$, and the condition for positivity thus is

$$(1 + \mu)\nu \leq 1 \,. \tag{37}$$

Note that this condition also implies other nice properties, such as the maximum principle $\min_j w_j^n \leq w_i^{n+1} \leq \max_j w_j^n$ and the total variation diminishing (TVD) property, see [60].

The flux-limited counterpart of the MOL scheme (22), which we studied earlier in ref. [41], can be obtained by taking the limit $\tau \to 0$ with $h > 0$ fixed, just as for the schemes without limiting. We then obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} w_i(t) = \frac{1}{h}(f_{i-1/2} - f_{i+1/2}) \,, \tag{38}$$

with fluxes as before from (34), (36), with $d_0 = 1/3$, $d_1 = 1/6$. The resulting MOL scheme gives nonnegative solutions for any $\mu > 0$ if the time-stepping is done exactly. Observe that the limiting procedure turns a linear difference scheme into a nonlinear one.

The free parameter μ from (36) has still to be chosen. Large values of μ give more accurate results, as more often the original flux will be obtained. It leads, for example, to less clipping of peaks. However, larger values for μ result in smaller allowable Courant numbers. In the MOL approach of ref. [41] it was found experimentally that $\mu = 1$ performs quite well and is more efficient than $\mu = 3$, for example. The fact that large values of μ are inefficient in the MOL approach is due to the ODE solvers. These require smaller step sizes, that is, smaller ν, to maintain positivity in the time integration if μ is increased. For example, with the explicit Euler scheme as ODE method, we get again condition (37) as a positivity condition imposed by explicit Euler (see also Sect. 4.4.1). In the following, we therefore consider $\mu = 1$ in the semidiscrete system (38). This specific limiter was first proposed by Koren. The limiter function is illustrated in Fig. 4.

With the DST scheme we have the flexibility to have μ depend on the Courant number. Taking μ as large as possible within the positivity constraint gives

$$\mu = (1 - \nu)/\nu \quad \text{for } 0 \leq \nu \leq 1 \,. \tag{39}$$

Now the DST scheme is positive for all $\nu \leq 1$ and thus the most accurate region for this scheme where ν is near 1 can be included. Moreover, for small Courant

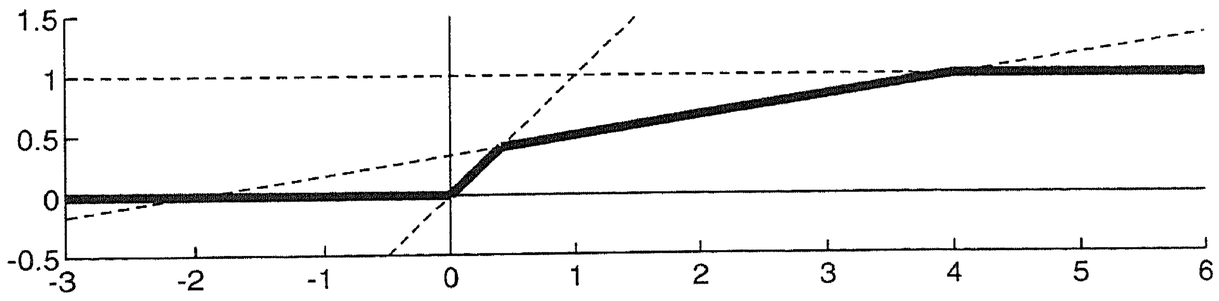


Fig. 4. Koren–Sweby limiter function for $\mu = 1$

numbers, μ gets large which is advantageous, e.g., less clipping of peaks compared to fixed $\mu = 1$ will be needed.

4.3 Related work and remarks

The choice for the third-order upwind-biased formulas is not primarily based on the third-order accuracy property. Compared to second-order central or fully upwind, the third-order formulas have much better shape-preserving properties (moderate oscillations and small phase errors). Moreover, in connection with flux limiting, the upwind-biased stencil is also a natural choice. A Sweby-type limiter already needs 3 points, so that the upwind stencil only needs one more. The complete stencil consisting of 5 points per direction thus offers a good compromise between computational efficiency, accuracy, and shape preservation.

With a variable velocity $u = u(x, t)$, the formulas can be applied by using the local velocity $u_{i+1/2} \approx u(x_{i+1/2}, t)$ in the flux expression. If $u_{i+1/2} < 0$, the stencil $\{x_{i-1}, x_i, x_{i+1}\}$ should be reflected around the point $x_{i+1/2}$ to maintain the upwind character. We then get

$$f_{i+1/2} = u_{i+1/2}\Big(w_{i+1} + \psi\Big(\frac{1}{\theta_{i+1}}\Big)(w_i - w_{i+1})\Big)\,. \tag{40}$$

The results on positivity remain valid. For both the MOL and the DST scheme the order of consistency 3 decreases to 2, pointwise. This can be proven only when the original fluxes are used. Flux limiting itself might also lead to some order reduction, even when u is constant. However, this will only happen in regions where the solution is nonsmooth or near extrema. Elsewhere the θ_i will be close to 1, so there the limiter is not active. Consequently, the effect of limiting on the global accuracy consists mainly of some numerical diffusion near extrema and steep gradients where this might be needed to maintain monotonicity. Formal statements on the order of accuracy seem difficult due to the various switches in the limiter, but numerical tests [40: fig. 3] indicate convergence with order 2 approximately.

The DST scheme can be derived in various ways. In ref. [40] it was derived for non-constant velocities by interpolation of primitive functions, similar to the approach of Colella and Woodward [19]. Alternative derivations of the DST scheme are given by Bott [17] and Leonard [57] (QUICKEST), for example. Some other, popular flux limiters for fully discrete schemes can be found in refs. [60] and [115], expressed in terms of $\phi(\theta) = 2\psi(\theta)/(1 - \nu)$. Typical examples are the MUSCL limiter

$$\phi(\theta) = \max\big(0,\ \min\big(2,\ 2\theta,\ \tfrac{1}{2}(1+\theta)\big)\big)$$

of van Leer [55] and the "superbee" limiter

$$\phi(\theta) = \max(0,\ \min(1, 2\theta),\ \min(\theta, 2))$$

of Roe [73]. The MUSCL limiter gives more diffusion and clipping of peaks than our choice (36), whereas the superbee limiter gives roughly the same peak values as

(36) and slightly better results for advection of a square wave. However, superbee is more inaccurate for smooth solutions, showing a strong tendency to turn smooth curves into straight lines, see the figures presented in ref. [115]. Our limiter fits into the general formula (5.13) in ref. [115], where it was called target limiting since there is a target function, the nonlimited flux, that is preserved as much as possible.

An alternative to flux limiting is flux-corrected transport (FCT), originally due to Boris and Book [16] but nowadays commonly used in the version of Zalesak [114]. Here the high-order fluxes are mixed with first-order upwind (donor cell) fluxes. The FCT procedure, however, is more complicated from a computational point of view than flux limiting. An advantage of FCT is that it can be applied to any discretization, as long as it is in flux form.

As for semi-Lagrangian methods, it is easy to obtain a DST form that is unconditionally stable. Consider once more scheme (32). For the computation of the fluxes $f_{i+1/2}$ we can allow the stencil to vary with the Courant number. With large Courant numbers also the contribution to the flux of nonadjacent cells should be taken into account. In this way we achieve unconditional stability while maintaining explicitness and mass conservation of the scheme. As an example, first consider the flux for the donor cell scheme

$$\frac{\tau}{h} f_{i+1/2} = \begin{cases} \nu_{i+1/2} w_i & \text{if } u_{i+1/2} \geq 0, \\ -\nu_{i+1/2} w_{i+1} & \text{if } u_{i+1/2} < 0, \end{cases}$$

with $\nu_{i+1/2} = |u_{i+1/2}|\tau/h$ being the local Courant number at the cell boundary $x_{i+1/2}$. In this form the scheme is stable under the CFL (Courant–Friedrichs–Lewy) restriction $\max_i \nu_{i+1/2} \leq 1$. The stability restriction is avoided by taking

$$\frac{\tau}{h} f_{i+1/2} = \begin{cases} \tilde{\nu}_{i+1/2} w_{i-k} + (w_{i-k+1} + \cdots + w_i) & \text{if } u_{i+1/2} \geq 0, \\ -\tilde{\nu}_{i+1/2} w_{i+k+1} - (w_{i+1} + \cdots + w_{i+k}) & \text{if } u_{i+1/2} < 0, \end{cases}$$

where

$$k = k_{i+1/2} = \lfloor \nu_{i+1/2} \rfloor \quad \text{and} \quad \tilde{\nu}_{i+1/2} = \nu_{i+1/2} - k_{i+1/2},$$

and $\lfloor \nu \rfloor$ denotes the largest integer being smaller than or equal to ν. Inserting this into formula (32), we see that for constant u the same formula is applied as for Courant numbers smaller than or equal to 1, with a shift over k grid points. Therefore the scheme will be unconditionally stable (for constant u). Note that this is similar as with semi-Lagrangian methods, but due to the conservation form of the scheme we still have mass conservation. A scheme of this type was introduced by LeVeque [61] for nonlinear conservation laws. Finally, instead of the donor cell fluxes we can use the third-order fluxes with limiting. Then the terms $\tilde{\nu}_{i+1/2} w_{i-k}$ are replaced by their third-order counterparts. The full formulas can be found in ref. [42].

4.4 Time integration

We next address the time integration of the semidiscrete MOL schemes (22) and (38). We consider the first-order explicit Euler method, also known as forward Euler method, a second-order explicit Runge–Kutta method, also called explicit trapezoidal rule, and a second-order explicit linear multistep method derived from the second-order BDF. The forward Euler method is discussed only for illustrational purposes as the combination of this method with third-order upwind is unstable. Hence this combination is not advocated for real use. On the other hand, both second-order methods have shown to perform very well in different tests, see, e.g., [41, 104, 106]. Both will also appear as special cases of schemes discussed in Sect. 5. Observe that we restrict ourselves to explicit time stepping. At the end of this section a brief remark is made on the use of implicit methods for advection.

For convenience of notation we write the ODE system (38) in the nonlinear vector function form

$$\frac{\mathrm{d}}{\mathrm{d}t}w = F(w), \quad w = [\ldots, w_{i-1}, w_i, w_{i+1}, \ldots]^T, \; F_i(w) = \frac{f_{i-1/2} - f_{i+1/2}}{h} . \tag{41}$$

In the same way the linear, constant coefficient system (22) is written as

$$\frac{\mathrm{d}}{\mathrm{d}t}w = Dw, \quad (Dw)_i = \frac{u}{h}\Big(-\frac{1}{6}w_{i-2} + w_{i-1} - \frac{1}{2}w_i - \frac{1}{3}w_{i+1}\Big) . \tag{42}$$

4.4.1 Explicit Euler method

The most simple explicit integration method for general ODE systems $\dot{w} = F(w)$ is Euler's rule

$$w^{n+1} = w^n + \tau F(w^n) . \tag{43}$$

Similar as the DST scheme (29) (the fluxes are different), explicit Euler proceeds from time level to time level by using only one evaluation of F. Hence per time step the method is very cheap. However, it is only first-order consistent. This means that, at best, the global error $w^n - w(t_n) = O(\tau)$ for $\tau \to 0$ and $n \to \infty$ with $t_n = n\tau$ fixed. Therefore the accuracy is mostly considered too low. Moreover, when combined with third-order upwind, the method lacks stability.

Stability. To examine time step stability for semidiscrete MOL schemes, we again apply the Fourier–von Neumann technique. In spite of the linearity assumption and the restriction to constant coefficients and periodic boundary conditions, for many important PDE problems the reliability of this technique is beyond question [34, 72]. So we apply explicit Euler to the linear system (42), which yields the linear recurrence relation

$$w^{n+1} = (I + \tau D)w^n . \tag{44}$$

The Ansatz of Sect. 4.1.1 is now extended with the Ansatz that the Fourier mode

$$w_i^n = \hat{w}(0)\mathcal{P}^n \, \mathrm{e}^{\sigma\omega x_i}, \quad \sigma = \sqrt{-1} \, , \tag{45}$$

solves (44) like (28) solves (42). Note that the amplification factor $\mathcal{P}$ then must approximate the exponential $\mathrm{e}^{\tau\lambda}$, λ representing eigenvalues (27) of the difference operator D. Inserting the Fourier mode (45) shows that $\mathcal{P} = \mathcal{P}(\tau\lambda)$ such that

$$\mathcal{P}(\tau\lambda) = 1 + \tau\lambda \, . \tag{46}$$

Fourier–von Neumann stability now means that all Fourier modes represented on the grid are not amplified in time, that is, we must satisfy the inequality

$$|\mathcal{P}(\tau\lambda)| \leq 1 \quad \text{for all } -\pi \leq \xi \leq \pi \, . \tag{47}$$

This condition imposes a restriction on the values $\tau\lambda$ or, equivalently, on the CFL number $\nu = (\tau u)/h$. More precisely, the inequality reads

$$\left|1 - \tfrac{1}{3}\nu(\cos\xi - 1)^2 - \tfrac{1}{3}\sigma\nu\sin\xi(4 - \cos\xi)\right| \leq 1 \, . \tag{48}$$

If we fix ν and let $\xi \to 0$, this inequality is easily shown to be false, irrespective of the size of ν. Consequently, explicit Euler is not stable.

Positivity. Curiously, for the Koren–Sweby limiter shown in Fig. 4, explicit Euler is positive under the CFL condition

$$\nu = \tau u/h \leq 1/2 \, . \tag{49}$$

This follows immediately from the observation that for system (41), the Euler rule (43) is equivalent to formula (35), if we insert the MOL limiter function in the flux expressions. The analysis following this formula then applies and condition (37) just appears to be the positivity condition for explicit Euler. Inserting $\mu = 1$ yields (49). Although explicit Euler is not recommended for practice, this positivity result is of interest in its own as it can be used to prove positivity for the two second-order integration methods discussed next.

4.4.2 Explicit trapezoidal rule

Consider the 2-stage Runge–Kutta formula

$$w^{n+1} = w^n + \tfrac{1}{2}\tau F(w^n) + \tfrac{1}{2}\tau F(w^n + \tau F(w^n)) \, , \tag{50}$$

henceforth called RK2. It proceeds from time level t_n to t_{n+1} by two evaluations of F. This 2-stage formula is also called the explicit trapezoidal rule, as it is obtained from the well-known implicit trapezoidal rule

$$w^{n+1} = w^n + \tfrac{1}{2}\tau F(w^n) + \tfrac{1}{2}\tau F(w^{n+1}) \, ,$$

by replacing w^{n+1} in the right-hand side with the explicit Euler result $w^n + \tau F(w^n)$. RK2 has second-order consistency. Since in air pollution modelling only modest accuracies are required, we consider second-order in time sufficient.

Stability. Time step stability is analyzed in the same way as for the Euler scheme. We then arrive at condition (47), with the polynomial $\mathcal{P}$ now given by

$$\mathcal{P}(\tau\lambda) = 1 + \tau\lambda + \tfrac{1}{2}\tau^2\lambda^2 \,. \tag{51}$$

In the ODE literature, $\mathcal{P}$ is called the stability function. An easy way to determine a good approximation to the critical CFL number is to draw $(\tau\lambda(\xi))$-loci for different values of ν and to visually inspect for which ν we get intersection with the boundary locus of the RK2 stability region $\{ z \mid |\mathcal{P}(z)| \leq 1 \}$. Figure 5 shows this stability region and the locus for $\nu = 0.87$. This locus appears to be close to the critical one. Observe that the critical point of intersection lies near the imaginary axis. We thus conclude that RK2 is Fourier–von Neumann stable if

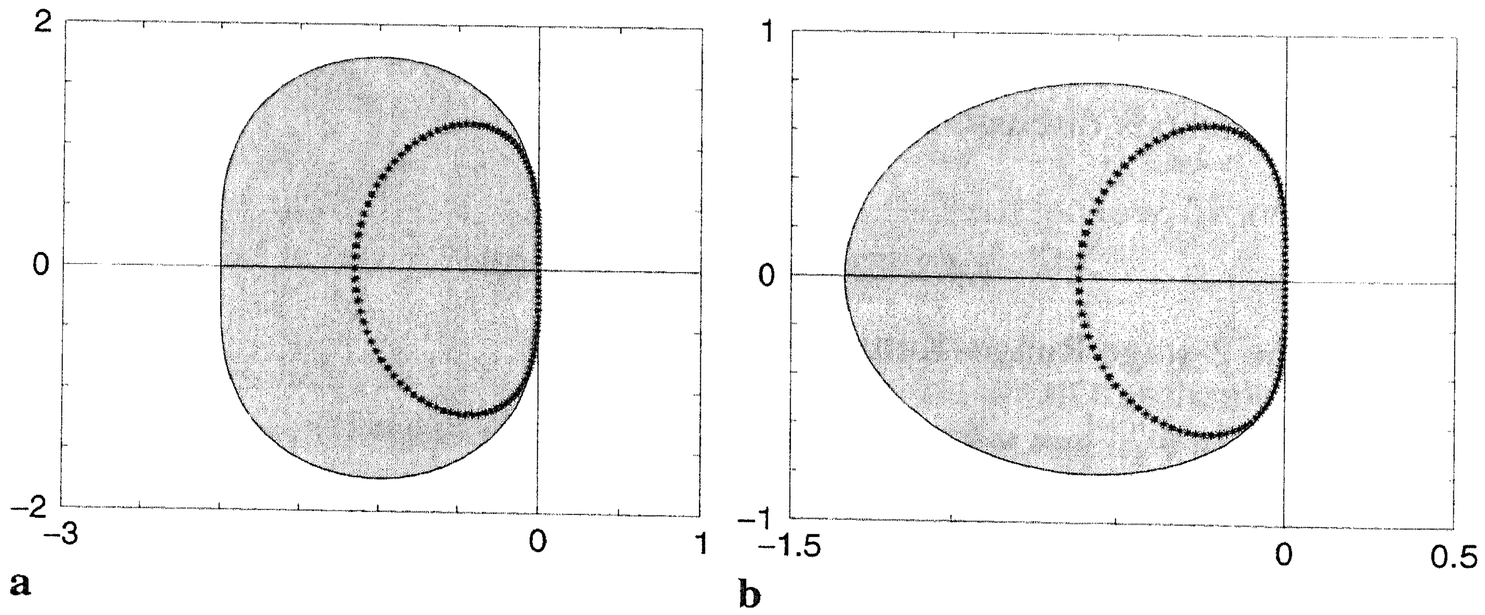
$$\nu = \tau u/h \leq 0.87 \,. \tag{52}$$

Positivity. To prove positivity for RK2 we rewrite (50) as

$$w_{n+1} = \tfrac{1}{2}w_n + \tfrac{1}{2}(\tilde{w} + \tau F(\tilde{w})), \quad \tilde{w} = w_n + \tau F(w_n) \,, \tag{53}$$

so that two successive explicit Euler steps appear, each with a step size τ. It follows immediately that if explicit Euler is positive, RK2 is positive too. So with flux limiting we have again positivity under the CFL condition

$$\nu = \tau u/h \leq 1/2 \,. \tag{54}$$



a **b**

Fig. 5. Stability regions (shaded) for RK2 (**a**) and EBDF2 (**b**) with the third-order spectrum for $\nu = 0.87$ (RK2) and $\nu = 0.46$ (EBDF2). Note the difference in scaling

Note that this condition is more restrictive than the stability condition (52). A systematic investigation on positivity of the general class of explicit Runge–Kutta methods is due to Shu and Osher [82]. It turns out that theoretical bounds guaranteeing positivity are quite restrictive. Many methods even fail to be positive irrespective of the size of the time step. However, experiments indicate that for most solution profiles a minor violation of the theoretical step size bound will not lead to negative solution values [41].

4.4.3 Explicit BDF2

A second integration formula that combines well with the third-order upwind discretization is the explicit backward-differentiation formula [104]

$$w^{n+1} = \tfrac{4}{3}w^n - \tfrac{1}{3}w^{n-1} + \tfrac{2}{3}\tau F(2w^n - w^{n-1}) , \tag{55}$$

henceforth called EBDF2. This 2-step formula is also second-order consistent and is obtained from the well-known implicit 2-step BDF

$$w^{n+1} = \tfrac{4}{3}w^n - \tfrac{1}{3}w^{n-1} + \tfrac{2}{3}\tau F(w^{n+1}) \tag{56}$$

by replacing w^{n+1} in the right-hand side with the extrapolated value $2w^n - w^{n-1}$. Because it is of 2-step type, an extra initial value is required to start the computation. For this purpose the explicit Euler rule can be used.

Stability. When combined with third-order upwind, EBDF2 can be shown to be Fourier–von Neumann stable under the CFL condition [104]

$$\tau u/h \leq 0.4617485908 \quad (\approx 0.46) . \tag{57}$$

The critical CFL number is about two times smaller than for RK2. On the other hand, one integration step with EBDF2 requires only one evaluation of F against two for RK2. The stability region is given in Fig. 5 with the $(\tau\lambda(\xi))$-locus for $\nu = 0.46$.

Positivity. EBDF2 can also be shown to be positive under a mild CFL restriction when flux limiting is used. We present a proof due to Hundsdorfer and van Loon (pers. commun., 1998). Like for RK2, this proof exploits the CFL condition on positivity for explicit Euler. Let $d^n = 2w^n - w^{n-1}$. Then (55) can be written as

$$d^{n+1} = \tfrac{2}{3}d^n + \tfrac{1}{3}w^n + \tfrac{4}{3}\tau F(d^n) = \tfrac{2}{3}(d^n + 2\tau F(d^n)) + \tfrac{1}{3}w^n . \tag{58}$$

Let $n = 1$ and suppose that the extra starting vector w^1 is computed by explicit Euler from a start vector $w^0 \geq 0$. Hence $w^1 = w^0 + \tau F(w^0)$ so that $d^1 = 2w^1 - w^0 = w^0 + 2\tau F(w^0)$. The positivity condition for explicit Euler then implies that $w^1 \geq 0$ and $d^1 \geq 0$ as long as

$$\tau u/h \leq 1/4 . \tag{59}$$

This also implies $d^1 + 2\tau F(d^1) \geq 0$ so that $d^2 \geq 0$. This in turn implies that $w^2 \geq \frac{1}{2}w^1 \geq 0$ which completes the proof for the first EBDF2 step using forward Euler as starting formula. The same reasoning holds for all successive steps, proving positivity under the CFL condition (59), which is again a factor two smaller when compared with RK2. Interestingly, the bound 1/4 can be sharpened to 5/16 if we substitute $w^n = \frac{1}{2}d^n + \frac{1}{2}w^{n-1}$ into (58), so that we get

$$d^{n+1} = \tfrac{5}{6}(d^n + \tfrac{8}{5}\tau F(d^n)) + \tfrac{1}{6}w^{n-1} .$$

However, we then must suppose additionally that the first step guarantees $d^1 \geq 0$ since with the bound 5/16 positivity of $d^1 = w^0 + 2\tau F(w^0)$ is not certain.

4.4.4 Implicit versus explicit time stepping

Explicit advection with a MOL scheme leads to CFL numbers between, approximately, 0.5 and 1.0. Nevertheless, in many practical computations explicit methods are considered efficient, as they allow a sufficiently large step size. In particular this is true for global and large-scale regional models. Such models use quite large grid boxes and very often temporally averaged wind fields (e.g., monthly), so that strong local winds are eliminated.

In small-scale regional or urban models, CFL limits may be felt more restrictive, due to finer grids and stronger winds. The CFL stability restriction can be avoided by using implicit time stepping. Implicit time steps are more costly because of the implicit relations that need to be solved. When limiting is used, the solution of the nonlinear implicit relations can become cumbersome. Another reason to be reluctant in advocating implicit methods is that they do not guarantee positivity for arbitrarily large step sizes, except for the implicit Euler method. Hundsdorfer [43] presents numerical results which show that even with quite moderate CFL numbers, this lack of positivity can result in notably negative values in regions with large gradients. These results question the use of implicit time stepping for positive scalar advection in atmospheric applications.

4.5 Multidimensional MOL approach

In applications we have to deal with multidimensional problems, say, in two dimensions with

$$c_t + (uc)_x + (vc)_y = 0 . \tag{60}$$

A clear advantage of the MOL approach is that the 1-D semidiscretization is naturally extended, by discretizing on a 2-D space grid each of the 1-D operators as before and adding the resulting expressions to obtain a 2-D semidiscrete system $\dot{w} = F(w)$. Variable velocities and boundary conditions render no particular difficulties either. See ref. [41] for details.

The integration of this 2-D system with schemes like RK2 or EBDF2 goes through

without any difficulty, as the schemes are explicit. We have to face larger Courant numbers, though. For problem (60) the expression becomes

$$\nu = \tau\left(\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y}\right) . \tag{61}$$

To determine critical step sizes τ, the bounds found in one dimension should be imposed on this inequality. In spite of the fact that a Fourier–von Neumann analysis is not applicable, for variable velocities the maximum of this expression over the grid often yields a reliable step size limit for stability. Positivity in the flux-limited case is guaranteed by using this maximized Courant number expression.

4.6 Multidimensional DST approach

4.6.1 Splitting

An easy way to extend the DST approach is dimension splitting [40, 60]. In two dimensions we then solve subsequently on $[t_n, t_{n+1}]$ the equations

$$c_t + (uc)_x = 0, \quad c_t + (vc)_y = 0 ,$$

with the DST scheme (32) and (36) to obtain

$$w_{ij}^* = w_{ij}^n + \frac{\tau}{\Delta x}\left(f_{i-1/2,j}^n - f_{i+1/2,j}^n\right) ,$$
$$w_{ij}^{n+1} = w_{ij}^* + \frac{\tau}{\Delta y}\left(g_{i,j-1/2}^* - g_{i,j+1/2}^*\right) .$$

The fluxes $f_{i+1/2,j}$ and $g_{i,j+1/2}$ are defined in x and y direction, respectively. It is well known that this simple splitting procedure is only first-order accurate, unless the wind field is constant, in which case there is no splitting error, see, for instance, [60, 49]. The order can easily be raised to two by using Strang splitting, which basically amounts to interchanging the order of the fractional steps after each complete time step of size τ. In a more general setting, Strang splitting is discussed in Sect. 5.2.

With regard to stability and positivity, splitting is advantageous compared to the otherwise more flexible MOL approach. From the 1-D considerations and the sequential nature of splitting, it follows immediately that positivity is guaranteed if

$$\max\left(\frac{\tau|u|}{\Delta x}, \frac{\tau|v|}{\Delta y}\right) \le 1 , \tag{62}$$

which should be satisfied over the whole grid. Recall that this is also the condition for stability, in the sense of Fourier–von Neumann, with the nonlimited DST scheme for constant velocities. Comparison with (61) shows that the multidimensional nature works out less restrictive.

4.6.2 Modified splitting

However, in spite of the positivity property, in more dimensions splitting may involve a lack of monotonicity. As pointed out by Bott [17], this can already occur if the sought concentration is constant in space and the wind field is divergence free and variable. The reason is that in the first split step the equation $c_t + (uc)_x = 0$ is approximated in a manner consistent with the conservation form. The intermediate solution is then positive but neither constant nor monotone. Bott [17] observed that first-order splitting may give qualitatively bad results with deformational flow fields with a background concentration. Experiments by Hundsdorfer and Spee [42] suggested that for advection on a plane much better results are obtained if one uses a genuine second-order splitting method. However, tests with advection on a sphere [42] revealed that the lack of monotonicity is also felt with second-order splitting. On the sphere the monotonicity problem is in a sense amplified by a singularity at the poles (see Eq. (63)).

To remedy this problem, a modification was suggested with the somewhat fancy name deformation correction [42]. This modification consists of multiplying, after each step, the concentrations w_{ij}^{n+1} with a certain factor, such that if we had started with $w_{ij}^n \equiv 1$, the concentrations w_{ij}^{n+1} would be the same as those obtained with the nonsplitted, donor-cell algorithm applied to this uniform concentration field. This modification yields indeed a notable improvement, but since it modifies concentrations, rather than fluxes, the resulting scheme is no longer strictly mass conservative. Although some tests [42] on an analytical wind field of Williamson and Rasch [110] gave results which were "almost" mass conserving, this point remained a matter of concern. A modification on the fluxes, with a similar goal, was suggested by Russell and Lerner [76]. Our attention was drawn to this paper by the article of Easter [24]. We will first describe this modification for "artificial densities". After that an adjustment for genuine meteo densities will be given.

If the wind field is divergence free, a constant concentration field at time t_n should still be constant at t_{n+1}. Suppose now that we are given velocities that are divergence free in the discrete form

$$\frac{1}{\Delta x}(u_{i+1/2,j} - u_{i-1/2,j}) + \frac{1}{\Delta y}(v_{i,j+1/2} - v_{i,j-1/2}) = 0 .$$

Then, if we introduce the artificial densities $\rho_{ij}^n \equiv 1$ at time t_n and set

$$\rho_{ij}^* = \rho_{ij}^n + \frac{\tau}{\Delta x}(u_{i+1/2,j} - u_{i-1/2,j}), \quad \rho_{ij}^{n+1} = \rho_{ij}^* + \frac{\tau}{\Delta y}(v_{i,j+1/2} - v_{i,j-1/2}) ,$$

we have $\rho_{ij}^{n+1} \equiv 1$ in spite of the fact that the intermediate results ρ_{ij}^* may give large variations. The splitting modification of Russell and Lerner now consists of calculating the fluxes not from the concentrations w_{ij}^n and w_{ij}^* but from the mixing ratios $q_{ij}^n = w_{ij}^n / \rho_{ij}^n$ and $q_{ij}^* = w_{ij}^* / \rho_{ij}^*$. The resulting scheme is

$$q_{ij}^n = \frac{w_{ij}^n}{\rho_{ij}^n}, \quad w_{ij}^* = w_{ij}^n + \frac{\tau}{\Delta x}\left(f_{i-1/2,j}^n - f_{i+1/2,j}^n\right),$$

$$q_{ij}^* = \frac{w_{ij}^*}{\rho_{ij}^*}, \quad w_{ij}^{n+1} = w_{ij}^* + \frac{\tau}{\Delta y}\left(g_{i,j-1/2}^* - g_{i,j+1/2}^*\right),$$

with mixing ratio fluxes $f_{i+1/2,j}$ and $g_{i,j+1/2}$. These fluxes are computed in the same way as the concentration fluxes, except that now the values q_{ij}^n and q_{ij}^* are used instead of w_{ij}^n and w_{ij}^*. Assume that the very first step of this algorithm gives $q_{ij}^n = 1$ for all i, j. Then the interpolation for the fluxes is trivial and we get $q_{ij}^* = 1$, since the formulas for w_{ij}^{n+1} and ρ_{ij}^{n+1} will be the same. In a similar way it follows that $q_{ij}^{n+1} = w_{ij}^{n+1}/\rho_{ij}^{n+1} \equiv 1$. Likewise, if we have a constant concentration field $w_{ij}^n \equiv C$, then it follows that $w_{ij}^{n+1} \equiv C$, a property not shared by the original splitting. Due to the fact that the intermediate quantities w_{ij}^* may be far from equilibrium, the interpolation in the second step may give large errors in the original splitting.

Provided these are available, one can also use meteo data for the densities. In that case the mixing ratio fluxes should be multiplied by density fluxes. Let $\rho_{i+1/2,j}$ and $\rho_{i,j+1/2}$ be the average density inflow during the time step at horizontal and vertical cell boundaries calculated by some preprocessor. In the above scheme, $f_{i+1/2,j}^n$ is then replaced by $\rho_{i+1/2,j} f_{i+1/2,j}^n$ and $g_{i,j+1/2}^*$ by $\rho_{i,j+1/2} g_{i,j+1/2}^*$. The above modified splitting procedure based on third-order fluxes produced very favorable results when tested by Petersen et al. [68], also in 3 dimensions. In that paper a convergence proof is presented for a simplified scheme based on the donor cell algorithm.

Finally we note that another splitting modification with the same objectives as above was presented by Lin and Rood [62]. We have no experience with this modification, but results of Lin and Rod [62] indicate that it also works well. Similar modifications were considered by Leonard et al. [58].

4.6.3 Two-dimensional DST scheme

A generalization of the DST formula (32) to two dimensions in conservation form, with an FCT procedure for limiting, was considered by Rasch [70]. The resulting formula is very complicated, even without limiting, and not very attractive from a computational point of view. Moreover, a generalization to three dimensions seems hardly feasible with this approach.

4.6.4 Remarks on semi-Lagrangian technique

A semi-Lagrangian method uses the method of characteristics to compute discrete solutions on a priori chosen space grids, as in the Eulerian grid approach. In one

dimension semi-Lagrangian methods are based on the advective form $c_t + uc_x = 0$. This equation shows that c is constant along characteristics $(x(t), t)$ given by the differential equation $\dot{x}(t) = u(x(t), t)$. Thus to find the value at (t_{n+1}, x_i), we can trace the characteristic passing through this point backward to the time level t_n, where it will pass through some point $(t_n, \bar{x}_i)$. The value at this point can be found by interpolation of the known w_j^n from the grid. The interpolation thus maps the Lagrangian solution back to the Eulerian grid after each time step. This technique carries over to multidimensional problems with variable wind fields.

Different monotone interpolations of Hermite type were considered by Williamson and Rasch [110] and Smolarkiewicz and Rasch [86]. The big advantage of semi-Langrangian methods is the absence of a Courant restriction. With large time steps it is only the calculation of the characteristics that needs care. However, a major disadvantage of such formulas is the lack of mass conservation. In particular for long time integrations this can be a deficiency. The cause of the lack of mass conservation lies in the fact that the methods are not in conservation form.

4.7 Numerical illustration

4.7.1 Molenkamp–Crowley problem

For a numerical illustration we solve the two-dimensional Molenkamp–Crowley test problem

$$c_t + (uc)_x + (vc)_y = 0, \quad t > 0, \, \Omega = [0, 1]^2 \, ,$$

defined by the divergence-free velocity field

$$u(x, y) = 2\pi(y - \tfrac{1}{2}), \quad v(x, y) = -2\pi(x - \tfrac{1}{2}) \, .$$

The characteristics $(x(t), y(t))$ are given by

$$x(t) = a \sin(2\pi t) + b \cos(2\pi t) + \tfrac{1}{2}, \quad y(t) = -b \sin(2\pi t) + a \cos(2\pi t) + \tfrac{1}{2} \, ,$$

a and b arbitrary constants, being circles with center $(1/2, 1/2)$. The velocity field thus defines a rotation with period 1 around the center of the domain. The velocity signs show that the rotation is clockwise. As solution we rotate a cylinder and cone of height 1, both with base radius 0.15 and centered at $(1/2, 3/4)$ and $(1/2, 1/4)$ at $t = 0$, respectively.

This test problem gives a good indication of the capability of advection schemes, since it involves variable Courant numbers and diagonal flow, and both the cylinder and the cone provide a numerically difficult solution profile. These profiles cannot be resolved with standard advection schemes if wiggles or negative solutions are not allowed. On the other hand, it is of course only a simplified model. For example, the modification on the dimensional splitting for DST using ρ_{ij}^* values is trivial here.

4.7.2 Test details and results

The numerical tests were performed on a cell-centered 50×50 grid with the flux-limited MOL scheme, using RK2 and EBDF2 for the time integration, and with the flux-limited DST scheme, the latter with dimensional splitting. At the inflow boundaries the fluxes are prescribed. For points adjacent to the outflow boundaries, where the schemes need additional points outside the domain, these additional values are provided by constant extrapolation from the interior.

The step size τ was chosen as $1/(100k)$, with integer k such that the numerical results for $\tau = 1/(100(k-1))$ were unstable. This resulted in

$$\tau = 1/400 \text{ for EBDF2}, \quad \tau = 1/300 \text{ for RK2}, \quad \tau = 1/200 \text{ for DST} .$$

Note that EBDF2 thus uses the maximal 2-D Courant number

$$\frac{\tau}{h}(|u|_{\max} + |v|_{\max}) = \frac{1}{4}\pi ,$$

which is larger than the previously given theoretical bounds for stability and positivity. We still get a stable, positive solution due to the fact that the largest Courant numbers are found at the corners of the domain where boundary conditions are dominant. Likewise, RK2 uses the maximal 2-D Courant number $\pi/3$, which is also too large. DST is applied with the Courant number $\frac{\tau}{h}\max(|u|_{\max}, |v|_{\max}) = \frac{1}{4}\pi$, which is within the range of the theoretical bound. To make the comparison of the results easier, the unconditional stability option for DST was not used here. That option seems more suited for problems where large Courant numbers arise away from the boundaries, for example, near the poles with advection on a sphere.

Numerical solutions are shown for EBDF2 and DST after 1, 2, and 5 full rotations. The pictures for RK2 were nearly identical to those of EBDF2 and are therefore omitted here. Figure 6a and b gives contour plots, with solid contour lines at 0.1, 0.3, 0.5, 0.7, 0.9 and dotted contour lines at 0.2, 0.4, 0.6, 0.8 and at 0.01 and $1 - 0.01$. Figure 7a and b contains the corresponding 3-D plots. The pictures for EBDF2 and DST look similar, showing good shape preservation and phase speed. Of course, sharp profiles like cylinders and cones are diffused due to upwinding and flux-limiting. The results for DST are somewhat less diffusive than those of EBDF2 and RK2. This is due to the underlying 1-D scheme, see the discussion at the end of Sect. 4.1.2. The artificial diffusion notably decreases upon grid refinement. Although one step with DST requires slightly more work than with EBDF2 since the coefficients depend on the Courant numbers, this is more than balanced by the fact that DST allows larger step sizes. So, for this test the results for DST are slightly better than for EBDF2 and RK2. If the spatial grid size is reduced, the three schemes become equally accurate.

The same conclusion was reached by Petersen et al. [68] with more challenging tests for a solid body rotation on a sphere and a 3-D real-life Radon problem. In that paper, the MOL scheme with RK2 as time integrator was used on a so-called reduced grid (see Fig. 8). DST was used both on a uniform grid (but with the unconditional stability option) and on the reduced grid with the usual Courant restriction. In those

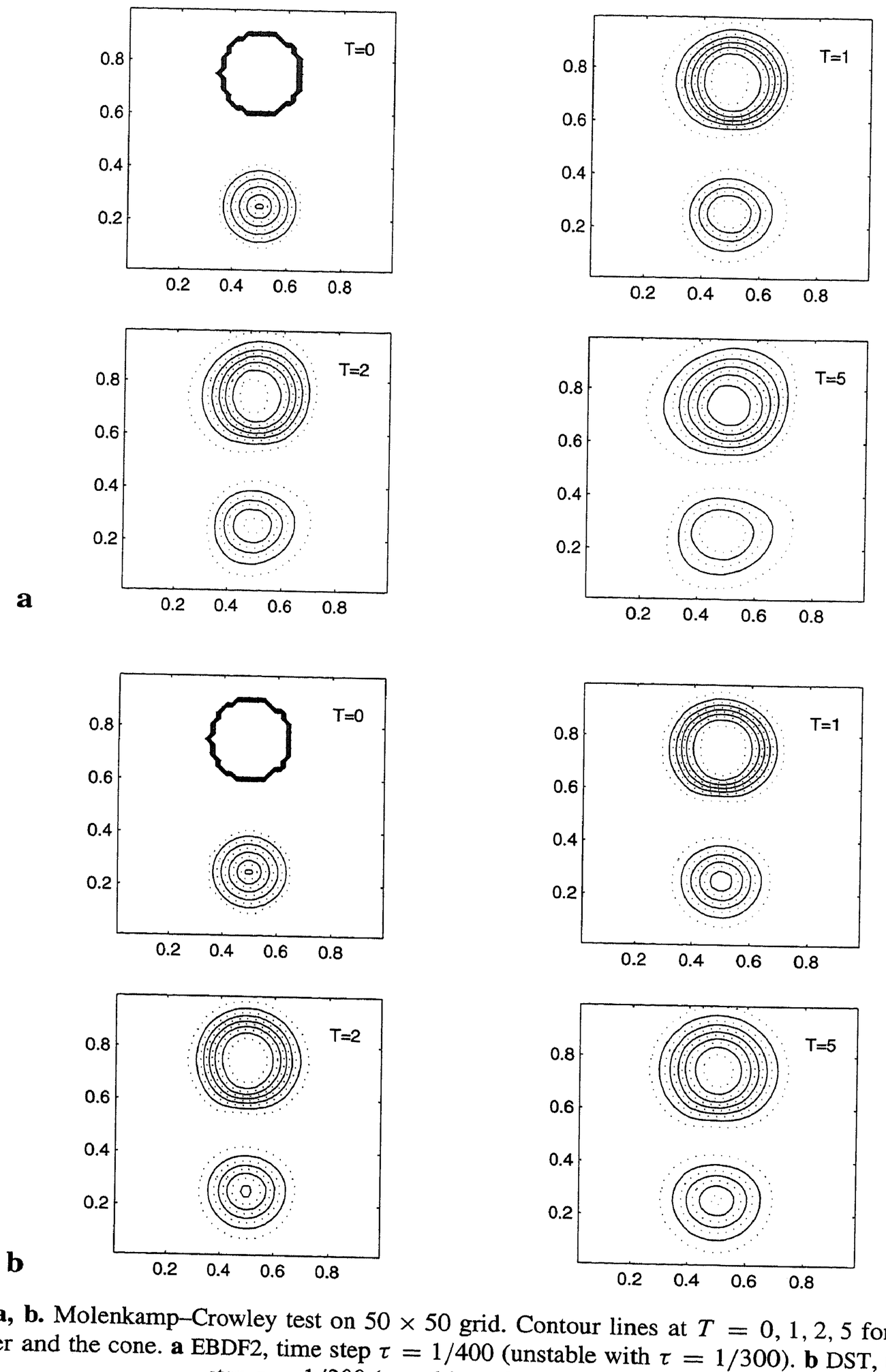
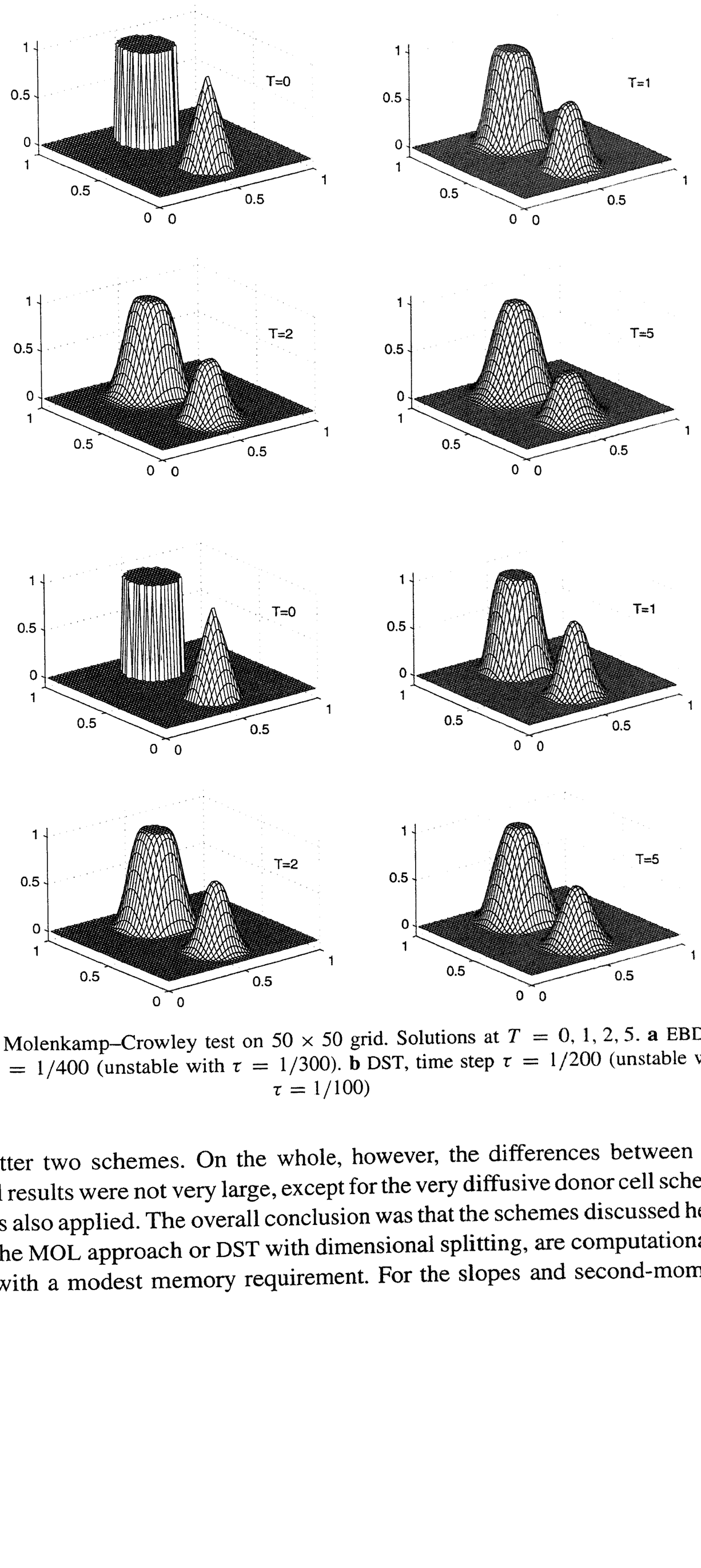


Fig. 6 a, b. Molenkamp–Crowley test on 50×50 grid. Contour lines at $T = 0, 1, 2, 5$ for the cylinder and the cone. **a** EBDF2, time step $\tau = 1/400$ (unstable with $\tau = 1/300$). **b** DST, time step $\tau = 1/200$ (unstable with $\tau = 1/100$)

tests also the slopes scheme of Russell and Lerner [76] and the second-moment scheme of Prather [69] were incorporated. Due to memory use and computational effort on fine grids, the conclusion by Petersen et al. [68] was somewhat unfavorable



a

b

Fig. 7 a, b. Molenkamp–Crowley test on 50×50 grid. Solutions at $T = 0, 1, 2, 5$. **a** EBDF2, time step $\tau = 1/400$ (unstable with $\tau = 1/300$). **b** DST, time step $\tau = 1/200$ (unstable with $\tau = 1/100$)

for the latter two schemes. On the whole, however, the differences between the numerical results were not very large, except for the very diffusive donor cell scheme which was also applied. The overall conclusion was that the schemes discussed here, either in the MOL approach or DST with dimensional splitting, are computationally efficient with a modest memory requirement. For the slopes and second-moment

schemes the memory requirement can become a problem when very fine grids are needed in global modelling, which is anticipated for the near future.

4.8 Final remarks on advection

In spite of its simplicity, the constant-coefficient 1-D model $c_t + uc_x = 0$ reveals major difficulties encountered in scalar numerical advection (stability, positivity, and monotonicity, shape preservation). Extending 1-D schemes to multidimensional problems $c_t + \nabla \cdot (\underline{u}c) = 0$, defined with a variable velocity field $\underline{u}$ and subjected to boundary conditions is usually possible. However, in applications one always encounters technicalities which are sometimes far from trivial. Four important issues found in real air pollution models are spherical coordinates, the pole problem, vertical wind, and orography.

Spherical coordinates are normally used in large regional and global models. The 2-D spherical-advection problem reads

$$\frac{\partial c}{\partial t} + \frac{1}{a \cos y}[(uc)_x + (vc \cos y)_y] = 0 \,, \tag{63}$$

where $x \in [0, 2\pi]$ denotes longitude, $y \in [-\pi/2, +\pi/2]$ latitude, and a is the radius of the earth. Cartesian grid upwind discretizations are easily reformulated in spherical coordinates. For example, the cell-centered MOL scheme now reads:

$$\frac{\mathrm{d}}{\mathrm{d}t} w_{ij} + \frac{1}{a \cos y_j}\Big[\frac{f_{i+1/2,j} - f_{i-1/2,j}}{\Delta x} + \frac{\cos y_{j+1/2} f_{i,j+1/2} - \cos y_{j-1/2} f_{i,j-1/2}}{\Delta y}\Big] = 0 \,. \tag{64}$$

The flux expressions are similar as before, except that they take into account the sign of the velocity vector. For example,

$$f_{i+1/2,j} = \max(u_{i+1/2,j}, 0) f^{+}_{i+1/2,j} + \min(u_{i+1/2,j}, 0) f^{-}_{i+1/2,j} \,,$$

where

$$f^{+}_{i+1/2,j} = w_{ij} + \psi(\theta_{ij})(w_{i+1,j} - w_{ij})$$

and, cf. (40),

$$f^{-}_{i+1/2,j} = w_{i+1,j} + \psi\Big(\frac{1}{\theta_{i+1,j}}\Big)(w_{ij} - w_{i+1,j}) \,.$$

The spherical coordinates give rise to a singularity at the poles. Problem (63) is not defined at the poles, and when approaching the poles, the longitudinal grid distances $\Delta x \cos y$ approach zero. This results in very small longitudinal CFL numbers causing a severe stability restriction on the step size of explicit advection schemes. Numerical remedies to this pole problem were surveyed by Williamson [109]. For scalar advection the reduced-grid approach provides a satisfactory solution. We adopted this approach in ref. [12] and in later papers on global numerical models

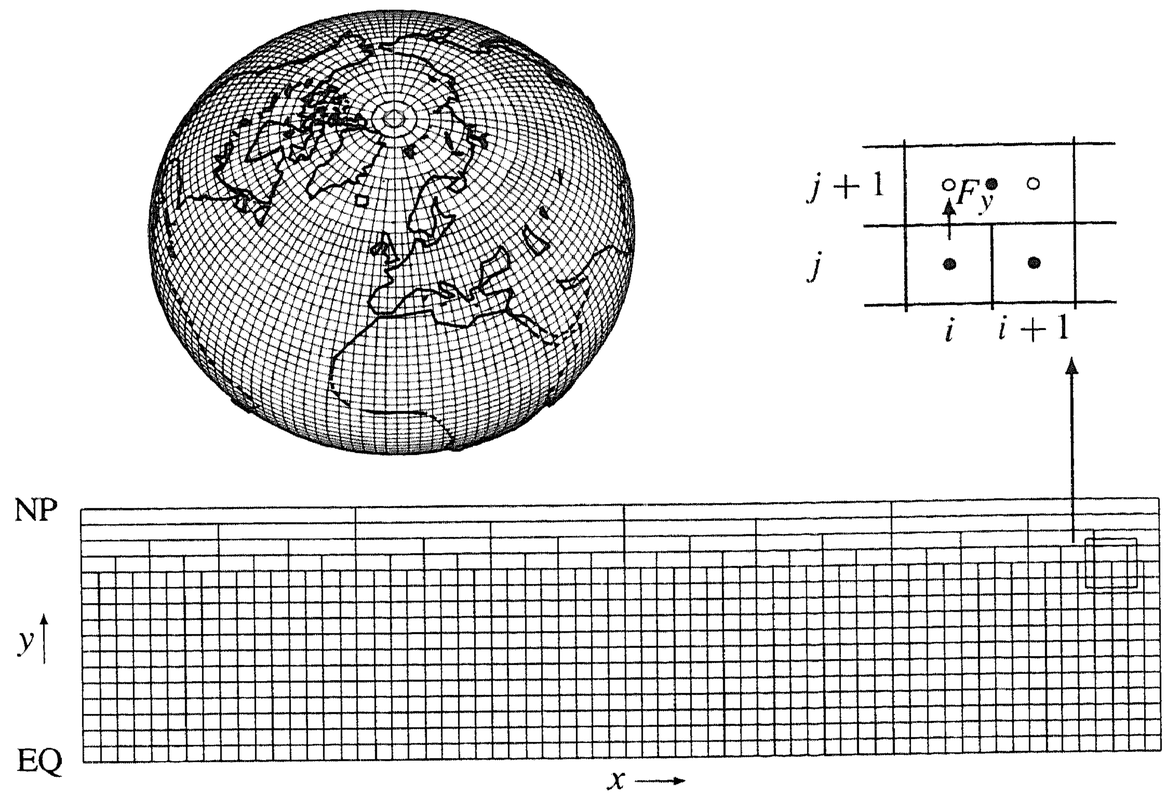


Fig. 8. A reduced 64 × 32 grid on the globe with its northern hemisphere part in longitude-latitude coordinates

collected in ref. [88]. A reduced grid coarsens the grid near the poles in a manner illustrated in Fig. 8. Grid reduction leads to a few technicalities at the grid interfaces, see refs. [12] and [88].

In nature the main transport by wind takes place parallel to the earth's surface. In this respect vertical advection may be neglected. However, in real models vertical advection is usually accounted for. A practical reason is that wind fields are often based on a combination of measurements and output from weather forecast or circulation models. Such fields are numerically adjusted on a priori prescribed 3-D grids to assure mass conservation, which may introduce vertical wind components. Allen et al. [2], who discussed a similar upwind scheme (Van Leer type) as the present ones, emphasized that the vertical-transport calculation can be sensitive to numerical adjustment.

In real models the vertical-wind calculation is further complicated by the orography of the earth. To model orography, modern models often use a coordinate transformation based on a hybrid coordinate system, following the orography at the surface and equal-pressure levels at the top. This hybrid sigma-pressure transformation, and vertical wind components, give rise to different technicalities. Those of the transformation itself can be found in ref. [11]. We again refer to Allen et al. [2] for a discussion on the vertical-wind treatment.

5 Solving the 3-D transport-chemistry problem

So far we have considered the numerical solution of advection and chemistry on its own. The next major research question is, how to efficiently solve the 3-D problem

where advection and chemistry are coupled, together with diffusion. In view of the many species and grid points, its numerical time integration requires a tailored approach so as to keep CPU times within reasonable bounds. In this section we will review three different possibilities enabling this, viz.,

- operator or time splitting, traditionally the most popular approach in the atmospheric modelling field since the paper by McRae et al. [65];
- implicit-explicit (IMEX) time stepping, which might be interpreted as splitting within a method rather than at the operator level;
- approximate matrix factorization, which might be interpreted as splitting at the level of the numerical algebra, rather than within the method or at the operator level.

5.1 The semidiscrete formulation

To facilitate a unified presentation, we discuss the three approaches for the semidiscrete system

$$\dot{w} = F(w) \,, \tag{65}$$

defined by the spatial discretization of

$$\partial c/\partial t + \nabla \cdot (\underline{u}c) = (K c_z)_z + R(c), \quad c = c(\underline{x}, t),\, c \in \mathbf{R}^m,\, \underline{x} \in \Omega \subset \mathbf{R}^3 \,. \tag{66}$$

Hence, $w(t)$ now represents the 3-D grid function approximating the true solution $c(\underline{x}, t)$ on some 3-D grid covering Ω. The vector function F is the complete semidiscrete operator, containing terms from advection, vertical diffusion, and chemistry. Note that we neglect horizontal diffusion. In real models, horizontal diffusion is often omitted, as it is minor compared to horizontal advection. Adding horizontal diffusion to codes solving (66) is straightforward, as an explicit integration will do. Horizontal diffusion can, for example, be combined with explicit horizontal advection. We further observe that for (66) one may also read its counterpart in spherical coordinates.

It should be emphasized that the semidiscrete formulation poses a restriction for the direct space-time advection scheme DST. This scheme does not fit in the MOL approach (see Sect. 4). DST can of course be used for (66), provided operator splitting is used. When appropriate we will return to this issue.

For the discussion to follow, we next split F in three parts, i.e., we rewrite (65) as

$$\dot{w} = F(w) \equiv F_{\mathrm{A}}(w) + F_{\mathrm{D}}(w) + F_{\mathrm{R}}(w) \,. \tag{67}$$

$\dot{w} = F_{\mathrm{A}}(w)$ is the nonlinear semidiscrete advection problem based on the mass-conservative, flux-limited, third-order upwind discretization. Eventually it may be extended with a horizontal diffusion discretization. Of practical interest is that $\dot{w} = F_{\mathrm{A}}(w)$ decouples into m subsystems, one for each species. When considered on its own, each subsystem can be integrated explicitly with simple second-order methods like RK2 or EBDF2 (see Sect. 4.4).

$\dot{w} = F_{\mathrm{D}}(w)$ is the linear semidiscrete system obtained from a mass-conservative

discretization of $(Kc_z)_z$ on a central 3-point stencil. This system usually requires implicit time stepping because vertical (turbulent) transport can be considerable and grid sizes in the vertical direction can be quite small. Of practical interest is that also $\dot{w} = F_D(w)$ decouples into m subsystems, one for each species. Further, each subsystem decouples over the horizontal grid. This means that we only encounter tridiagonal implicitness, which is cheap, irrespective of the choice of implicit integrator.

$\dot{w} = F_R(w)$ contains chemistry terms. Hence this system is stiff and requires implicit time stepping or a special-purpose method as discussed in Sect. 3.2. Of importance is that $\dot{w} = F_R(w)$ is decoupled over the grid, but coupled over the species. Emission and deposition terms associated to chemistry are also put in $F_R(w)$.

To keep the presentation general, we do not specify boundary conditions for (66). Instead, semidiscrete boundary conditions of various kinds are supposed to be included in F_A or F_D. For example, at the upper side of the domain Ω one might encounter a no-flux condition for the vertical-diffusion process. This condition is then placed in F_D. Likewise, surface emissions might be incorporated as a Dirichlet boundary condition for the vertical-diffusion process. At the surface one also encounters mixed flux conditions, containing surface emission and dry deposition. The incorporation of emission and deposition may differ per model. For example, surface emissions are sometimes treated as fluxes (boundary condition) but also as source terms directly linked with chemistry.

When assembling a time integration method for the 3-D system (66), one should again take into account the properties of mass conservation and positivity. The original PDE problem (66) conserves mass spatially through the advective- and diffusive-transport terms $\nabla \cdot (\underline{u}c)$ and $(Kc_z)_z$. This property is maintained in the semidiscretization by using difference formulas in the flux form. Standard Runge–Kutta and linear multistep methods retain this property. As explained in Sect. 3, molecular mass is conserved in the chemistry model $R(c)$. Implicit and linearly implicit integration methods conserve mass when they work with the true analytic Jacobian matrix. The special-purpose methods from Sect. 3.2 do not. For positivity similar remarks can be made. It should be emphasized, however, that truly retaining positivity in the chemistry integration is troublesome. See Sect. 3.

Step size notation. In the remainder we switch to the notation Δt for a temporal step size. Likewise, t_s denotes a discrete time point such that $t_{s+1} = t_s + \Delta t$. The step size Δt thus may represent a step size used by an operator-splitting scheme and $[t_s, t_{s+1}]$ a temporal split interval.

5.2 Operator splitting

Operator splitting, or time splitting, is in vogue already for a long time. In the atmospheric modelling field it is the standard way of solving the 3-D transport-chemistry problem since the paper by McRae et al. [65]. The basic idea of operator splitting is to treat processes like advection, diffusion, and chemistry on their own

in numerical time-stepping, so as to enable an easy use of well-prepared, tailored solvers for these different subprocesses.

5.2.1 Standard first-order splitting

For the semidiscrete system (67), the most simple operator splitting amounts to sequentially solving the three subproblems

$$\begin{aligned} \dot{w} &= F_{\mathrm{A}}(w), \quad t_s \le t \le t_{s+1} , \\ \dot{w} &= F_{\mathrm{D}}(w), \quad t_s \le t \le t_{s+1} , \\ \dot{w} &= F_{\mathrm{R}}(w), \quad t_s \le t \le t_{s+1} , \end{aligned} \tag{68}$$

over split intervals $[t_s, t_{s+1}]$, $s = 0, 1, 2, \ldots$, while using for each separate integration the result from the preceding integration as initial value. Because we do not directly solve $\dot{w} = F(w)$, a so-called splitting error is introduced. A formal Taylor series expansion of (68) shows that the original problem $\dot{w} = F(w)$ is approximated up to $O(\Delta t)$ in the split step size $\Delta t = t_{s+1} - t_s$. Hence the formal order of consistency is equal to one. It should be emphasized that in general this splitting error always exists, even when the subproblems are solved exactly. Only in the special case of commuting operators, splitting as above leaves no error (see Sect. 5.2.5). In the actual use of (68) one will of course also encounter numerical integration errors. These errors add to the splitting error [49].

Stability for (68) is determined by the stability properties of the subprocesses in view of the purely sequential nature. Likewise, when the numerical methods used for the subprocesses conserve mass and retain positivity, this holds for the complete approximation method.

5.2.2 Second-order Strang splitting

In an early influential paper, Strang [94] pointed out that the formal order $O(\Delta t)$ is raised to $O(\Delta t^2)$ if the computations are symmetrized around $t = t_{s+1/2}$. Since then Strang splitting is popular. The second-order symmetric version of (68) reads

$$\begin{aligned} \dot{w} &= F_{\mathrm{A}}(w), \quad t_s \le t \le t_{s+1/2} , \\ \dot{w} &= F_{\mathrm{D}}(w), \quad t_s \le t \le t_{s+1/2} , \\ \dot{w} &= F_{\mathrm{R}}(w), \quad t_s \le t \le t_{s+1} , \\ \dot{w} &= F_{\mathrm{D}}(w), \quad t_{s+1/2} \le t \le t_{s+1} , \\ \dot{w} &= F_{\mathrm{A}}(w), \quad t_{s+1/2} \le t \le t_{s+1} . \end{aligned} \tag{69}$$

As already mentioned, the splitting technique is very popular due to its simple concept and the freedom in choosing different solvers for advection, diffusion, and chemistry. In connection with advection it is emphasized that also the DST scheme can be used within operator splitting. It simply fits in the above splitting formulations if we replace the solution steps for the ODE system $\dot{w} = F_{\mathrm{A}}(w)$ by a fully discrete

relation like

$$w^{\text{new}} = \mathcal{A}\left(w^{\text{old}}\right) ,$$

where $\mathcal{A}$ denotes the mapping defined by the DST scheme used.

A fundamental question for any splitting application should be how to relate numerical errors at substeps with the $O(\Delta t)$ or $O(\Delta t^2)$ splitting error. It is clear that a certain balance should exist, since otherwise part of the computation would readily be too accurate or inaccurate, resulting in loss of efficiency. It appears that answering this question is difficult. Instead, in applications very often operational demands dictate choices. Off-line meteo data, such as wind fields for the advection and temperature and humidity for the chemistry, is refreshed for a priori fixed time intervals. This data is then often linearly interpolated from t_s to t_{s+1}, so as to secure a smoother evolution. The split step size Δt is mostly adjusted to the a priori fixed time points t_s and t_{s+1}. For advection one usually selects the maximal step size allowed by the CFL condition, such that it fits with (integer fractions of) Δt. The step size for diffusion is often chosen in a similar manner. Very often the chemistry is integrated with a variable step size governed by some form of local error control. Since this requires a start-up, it seems advantageous to integrate once over a distance Δt instead of twice over a distance $\frac{1}{2}\Delta t$, explaining why the chemistry is often put in the middle.

In applications, one encounters further splittings inside the ones discussed here. Even sinks and sources are splitted. It seems wise, though, to minimize the number of splitted operators. Simply because in general each single splitted operator will involve a splitting error and an additional numerical integration error. Because it is hard to find the best balance between many different errors, it is best to avoid them if possible.

5.2.3 First-order source splitting

In the splittings (68) and (69) chemistry alternates with advection and diffusion such that a discontinuity in the concentrations for the chemistry occurs at any time point t_s. In general, this discontinuity will result in stiff transients. Stiff solvers often exhibit difficulties in resolving stiff transients requiring smaller step sizes in transient phases. Hence, these discontinuities, being an artefact of the splitting, may increase the computational costs for the chemistry part.

An alternative splitting, aimed at circumventing solution discontinuities, is source splitting. Source splitting is advocated in refs. [46], [50], [89], [95], and [111]. The underlying idea is to treat the transport as a piecewise constant source over split intervals within the chemistry integration. Consider the semidiscrete system

$$\dot{w} = F(w) \equiv F_{\text{R}}(w) + F_{\text{T}}(w), \quad F_{\text{T}}(w) = F_{\text{A}}(w) + F_{\text{D}}(w) . \tag{70}$$

Source splitting amounts to approximate this system at successive split intervals by

$$\dot{w} = \tilde{F}(w) \equiv F_{\text{R}}(w) + \tilde{F}_{\text{T}}(w(t_s)), \quad t_s \le t \le t_{s+1} , \tag{71}$$

where the operator $\tilde{F}_{\mathrm{T}}$ represents a temporal approximation to the semidiscrete transport operator F_{T}, kept constant over each of the split intervals. Because the source is piecewise constant over intervals of length Δt, the splitting error here must be $O(\Delta t)$ in general.

One has basically the same freedom as in (68) or (69) for one's favorite combination of transport algorithms. Suppose that the transport problem is integrated over $[t_s, t_{s+1}]$, yielding the result $\tilde{w}^{s+1}$ for the start vector w^s. We then define

$$\tilde{F}_{\mathrm{T}}(w^s) = \frac{\tilde{w}^{s+1} - w^s}{\Delta t} \,. \tag{72}$$

In the actual application one has of course to reckon with numerical stability, similar as in the substeps of Strang splitting. Necessary is stability for the transport computation alone and whether this is sufficient for the whole will depend on the chosen combination. A particular scheme which for advection underlies RK2 and for diffusion a Rosenbrock method, will be presented in Sect. 5.4.

If the transport part slowly varies in time, source splitting is probably an attractive option. Note that with exact integration any source-splitting scheme applied to the inhomogeneous constant-coefficient linear system $\dot{w} = Aw + C$ with C representing a constant source, yields the true solution

$$w^{s+1} = \mathrm{e}^{\Delta t A}\left(w^s + (I - \mathrm{e}^{-\Delta t A})A^{-1}C\right) \,.$$

On the other hand, both splittings (68) and (69) yield a splitting error in this trivial case. For example, solving $\dot{w} = C$ first and then $\dot{w} = Aw$ would yield

$$w^{s+1} = \mathrm{e}^{\Delta t A}(w^s + \Delta t C) \,.$$

Compared to Strang splitting, a disadvantage of source splitting may be the $O(\Delta t)$ splitting error. Yet we believe that with respect to accuracy, source splitting can be of interest. Because when a stiff problem is involved, the second-order in Strang splitting may not show up unless Δt is unrealistically small. Following Sportisse [91], in the next section we will numerically illustrate this, again for an academic but otherwise instructive linear example problem.

5.2.4 Numerical example

Consider the linear ODE problem

$$\dot{w} = Aw + Bw, \quad t > 0 \,, \tag{73}$$

for which the constant matrix A is stiff and the constant matrix B is nonstiff, relative to a step size Δt. By this we mean that $||\Delta t A|| \gg 1$ and $||\Delta t B|| < 1$. Assuming exact integration, for this linear problem we consider the first-order splittings (see (68))

$$\dot{w} = Aw, \quad \dot{w} = Bw \quad \text{(called AB)} \,,$$
$$\dot{w} = Bw, \quad \dot{w} = Aw \quad \text{(called BA)} \,,$$

the second-order Strang splittings (see (69))

$$\dot{w} = Aw, \quad \dot{w} = Bw, \quad \dot{w} = Aw \quad \text{(called ABA)},$$
$$\dot{w} = Bw, \quad \dot{w} = Aw, \quad \dot{w} = Bw \quad \text{(called BAB)},$$

and the source splitting

$$\dot{w} = Aw + \left(e^{\Delta t B} - I\right)(\Delta t)^{-1} w(t_s) \quad \text{(called ST)}.$$

Note that BA and AB only differ in the order "nonstiff followed by stiff" and "stiff followed by nonstiff", respectively. We also have this difference in order for ABA and BAB.

We now choose A and B as

$$A = \begin{pmatrix} -1 & \varepsilon^{-1} \\ 0 & -\varepsilon^{-1} \end{pmatrix}, \quad 0 < \varepsilon \ll 1, \quad B = \begin{pmatrix} -0.5 & 1 \\ 1 & -2 \end{pmatrix}.$$

With this definition A is stiff and B is nonstiff for step sizes Δt between ε and 1, approximately. If Δt gets smaller than ε, A is no longer stiff. As a test example, this linear model is therefore of interest only for step sizes $\Delta t > \varepsilon$. For $t > 0$, the solution $w_2(t)$ is in fact proportional to ε. We therefore call w_2 here the stiff component. Likewise, $w_1(t)$ is dominated by the exponential associated with the nonstiff eigenvalue. Therefore w_1 is here called the nonstiff component. We put $\varepsilon = 10^{-5}$ and $w(0) = [1, 1]^T$ and apply the five splittings over the time interval $[0, 1]$ using step sizes $\Delta t > \varepsilon$, i.e., in the stiff range. Figure 9 shows the relative errors in $w_1(1) \approx 1.2130$ and $w_2(1) \approx 0.1213 \cdot 10^{-4}$.

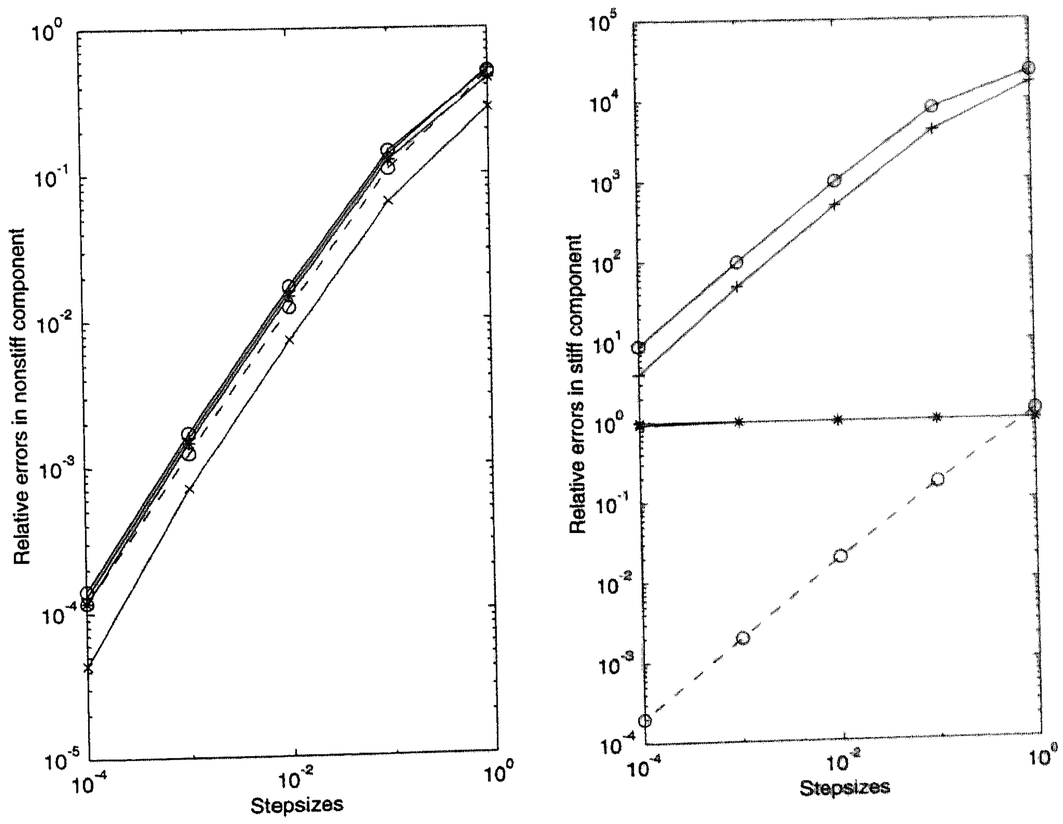


Fig. 9. Splitting errors for the linear example problem from Section 5.2.4. $-*-$, BA; $-\circ-$, AB; $-\times-$, ABA; $-+-$, BAB; $--\circ--$, ST splitting

The relative errors are quite revealing. For the nonstiff component they are almost equal for all splittings, except the Strang splitting ABA yields a slightly smaller error. Apparently this is due to the first-order convergence observed for all splittings, including ABA and BAB. What we here encounter for the Strang splittings ABA and BAB is order reduction from 2 to 1 in the stiff case. The reduction can be explained from the fact that in the stiff case, where $||\Delta t A|| \gg 1$, the standard Taylor expansion argument based on truncation of terms breaks down. In the stiff case the remainder terms in the series expansion cannot be neglected for arbitrary solutions.

For the stiff component we observe huge differences. Splittings AB and BAB show first order but the errors are large. Splittings BA and ABA do not converge but the errors are significantly smaller than for AB and BAB. What happens here is that AB and BAB do not mimic the exponential solution decay for the stiff component, resulting in these large relative errors. They do not decay, simply because the nonstiff solution step, indicated with B, has been put at the end of the split step. In this way there is an error-prone coupling from nonstiff to stiff solution components, resulting in a loss of decay for stiff components. On the other hand, splittings BA and ABA do have the stiff solution step at the end, with the consequence that the stiff component is made exponentially small. Since the true value is proportional to ε, we here encounter a relative error equal to 1 and no convergence as long as we are in the stiff step size range. Finally, the source splitting ST performs very satisfactorily for this linear example problem. It shows first-order convergence for both components so that the stiff component is computed much more accurately than in all the other cases.

For linear ODE problems, Sportisse [91] has analyzed this order reduction phenomenon with tools from singular-perturbation theory, amongst others proving first-order convergence for so-called reduced solutions. He has also presented a number of numerical examples, similar as the above, and a more realistic numerical test for a reaction-diffusion system based on nonlinear chemistry. This test confirms that putting the stiff computation at the end, as in BA and ABA, yields smaller relative errors. These results are interesting and warrant further research into the question how the splittings compare in real atmospheric applications. See, for example, Wolke and Knoth [111], who report promising results for a special source type splitting scheme implemented in a real meso-scale model.

Interestingly, the same order reduction phenomenon is observed when Runge–Kutta methods are applied to stiff, nonlinear ODEs (on B-convergence see [21: chap. 7]). It further occurs for PDE problems with time-dependent boundary conditions and source terms. See, e.g., ref. [39], where LOD (locally one-dimensional) splitting methods are discussed, and ref. [81], where it is shown that even explicit Runge–Kutta methods applied to hyperbolic problems may suffer from order reduction.

5.2.5 Commuting operators

From the theoretical point of view, the success of the splittings (68) and (69) is primarily determined by the splitting error introduced by solving the subproblems one after another in a completely decoupled manner. In general, this splitting error always exists, also when all subproblems are solved exactly. However, there do exist cases for which the splittings are exact. This occurs if the splitted operators are commuting. Linear operators, such as constant square matrices A and B, are said to commute if $AB = BA$. A consequence is that when splitting the ODE system

$$\dot{w} = Aw + Bw \ ,$$

as in (68) or (69), no splitting error results. This follows from the exponential solution operator, for which commutativity goes through, i.e.,

$$e^{tA+tB} = e^{tA}\, e^{tB} = e^{tB}\, e^{tA} \ .$$

Interestingly, the notion of commuting operators can be extended to the general nonlinear case, both in finite dimension (ODE case) and infinite dimension (PDE case). Specifically, the splittings (68) and (69) are exact if and only if all F_k and F_l commute with one another, i.e., for all w,

$$F_k'(w)F_l(w) = F_l'(w)F_k(w) \quad \text{(the primes denote differentiation to } w) \ . \tag{74}$$

This result can be proven by using the concept of Lie operator (see Lanser and Verwer [49] and references therein). Elaboration of the semidiscrete commutators $F_k'(w)F_l(w) - F_l'(w)F_k(w)$ is in general a very cumbersome task, however. For the PDE problem itself this is easier and in ref. [49] one finds an analysis directed at our 3-D transport-chemistry problem (1). For the autonomous, pure initial-value problem

$$\partial c/\partial t + \nabla \cdot (\underline{u}c) = \nabla \cdot (\underline{K}\nabla c) + R(c), \quad c = c(\underline{x}, t),\ c \in \mathbf{R}^m \ ,$$

it is proved that

- advection commutes with diffusion if $\underline{u}$ and $\underline{K}$ are independent of $\underline{x}$,
- advection commutes with chemistry if $\nabla \cdot \underline{u} = 0$ and R is independent of $\underline{x}$,
- diffusion commutes with chemistry if R is linear in c and independent of $\underline{x}$.

These results are of interest in themselves but, unfortunately, they do reveal that in the actual application of (68) or (69) the splitting error will always exist since the conditions for true commutativity are too strict. Also an explicit expression for the leading term of the local splitting error has been derived [49] by the Baker–Campbell–Hausdorff formula. The result is too complicated to provide much insight, though. Including boundary conditions in the analysis would even lead to a more complicated analysis. A few comments on this can also be found in ref. [49]. Finally we wish to emphasize that source splitting does not benefit from commuting operators. This type of splitting does yield a splitting error even when the operators commute.

5.3 Implicit-explicit methods

An alternative to the above splitting methods is provided by IMEX methods. The aim of an IMEX method is also to reduce somehow the implicitness in multidimensional PDE calculations, for example, to enable an explicit treatment of advection in advection-diffusion calculations as analyzed by Ascher et al. [3, 4].

5.3.1 IMEX-BDF2

Here we focus on a method of the linear multistep type, viz., a variant of the 2-step BDF2 method we encountered in Sects. 3.2.3 and 4.4.3. Applied to the semidiscrete system (67), this IMEX variant reads

$$w^{s+1} = \tfrac{4}{3}w^s - \tfrac{1}{3}w^{s-1} + \tfrac{2}{3}\Delta t\, F_{\mathrm{A}}(2w^s - w^{s-1}) + \\ + \tfrac{2}{3}\Delta t\big(F_{\mathrm{D}}(w^{s+1}) + F_{\mathrm{R}}(w^{s+1})\big)\,. \qquad (75)$$

It is obtained from its implicit counterpart by replacing the unknown w^{s+1} in F_{A} by the extrapolation $2w^s - w^{s-1}$, so that for advection alone (75) is identical to the explicit formula (55). For application to atmospheric transport-chemistry problems, this particular method has been examined by Verwer et al. [104]. In that paper one also finds references to related methods used in atmospheric applications, e.g., refs. [50] and [53]. IMEX methods of Runge–Kutta type for time-dependent PDEs were analyzed by Ascher et al. [4]. Knoth and Wolke [52] discussed an application towards atmospheric problems of a particular Runge–Kutta type IMEX method.

A difference with operator splitting is that one and the same step size Δt is used for advection, diffusion, and chemistry. Since the extrapolation $2w^s - w^{s-1}$ does not lower the second-order consistency, for air pollution models method (75) is believed to provide sufficient accuracy. With this method we stay close to a solid stiff ODE method, which is considered to be advantageous. A disadvantage of the IMEX approach, compared to operator splitting, is that only the advection integration is made easier. Diffusion and stiff chemistry are still implicitly coupled. In this sense the IMEX approach provides less flexibility. The question how to efficiently solve diffusion and stiff chemistry in air pollution models is addressed in Sect. 5.3.3.

The major impact of the extrapolation is a stability change. While the implicit method is L-stable, it is obvious that with the IMEX method we have to settle for less. A comprehensive linear stability analysis (basically Fourier–von Neumann) for (75) and similar methods has been presented in ref. [29]. From that paper we quote the following interesting results.

5.3.2 Stability for IMEX-BDF2

Consider the scalar, complex-valued test model

$$\dot{w} = \lambda_{\mathrm{A}} w + \lambda_{\mathrm{I}} w\,, \qquad (76)$$

in which λ_A and λ_I are supposed to represent eigenvalues for linear operators F_A and $F_D + F_R$, respectively. For example, if we diagonalize the chemistry part Mc in the linear constant-coefficient PDE system $c_t + uc_x + vc_y + wc_z = Kc_{zz} + Mc$, spatially discretize on a uniform grid and decompose the resulting linear system in Fourier modes, we end up with a scalar equation that fits in (76). Hence Fourier–von Neumann stability for this linear PDE system can be examined through stability for the test model. Assuming third-order upwind discretization for advection, λ_A is just the sum of three different complex eigenvalues (27). Likewise, λ_I is the sum of a real eigenvalue coming from Kc_{zz} and an eigenvalue from M, possibly complex valued. In the stability analysis the eigenvalues are supposed to have a nonpositive real part.

Application of (75) to (76) yields the recurrence relation

$$(1 - z_I)w^{s+1} = \left(\tfrac{4}{3} + \tfrac{4}{3}z_A\right)w^s - \left(\tfrac{1}{3} + \tfrac{2}{3}z_A\right)w^{s-1} ,$$

where $z_I = \Delta t \lambda_I$ and $z_A = \Delta t \lambda_A$. For any given pair (z_A, z_I), stability for evolving time now demands that the sequence $\{w^s\}$ is bounded uniformly in s. A slightly stricter condition is $w^s \to 0$ for $s \to \infty$. The latter is true if the two characteristic roots of the recurrence relation lie inside the unit circle. The stability analysis thus amounts to determine the set of (z_A, z_I)-values for which this holds. The resulting set is called the (interior) stability region of the recurrence relation. Trivially, for $z_I = 0$ the explicit eigenvalue z_A then must lie in the stability region of the explicit BDF2 method (55), while for $z_A = 0$ the implicit eigenvalue z_I must lie in the stability region of the original implicit BDF2 method, which is the left half complex plane in view of the L-stability property.

For the general case $z_A, z_I \neq 0$, the analysis appears to be far from trivial. In ref. [29] the two following questions are addressed.

Suppose that z_A lies in the stability region of the explicit method, denoted by $\mathcal{S}$. What restrictions are to be placed on the location of z_I to have stability?

Suppose that z_I lies in the left half-plane, included in the stability region of the implicit method. What restrictions are to be placed on the location of z_A to have stability?

For the current IMEX method the answers are shown in Fig. 10a and b. Figure 10a, associated to the first question, shows a smaller region for z_I pointing to L(α)-stability with an angle $\alpha < 90°$: z_I must lie in the wedge $\{ z_I \in \mathbf{C} \mid |\arg(-z_I)| < \alpha \}$ with $\alpha \approx 0.32\pi$. We have the impression that for atmospheric chemical-kinetics problems this eigenvalue restriction renders no serious limitation. This would mean that for our application the unconditional linear stability of the implicit BDF2 is maintained for the chemistry-diffusion part, so that the IMEX method is linearly stable as long as $z_A \in \mathcal{S}$.

Figure 10b shows the answer to the second question. It turns out that if we want to really maintain the L-stability property for the chemistry-diffusion part, the explicit eigenvalues z_A are restricted to a smaller region, which is denoted by $\mathcal{D}$. We see that $\mathcal{D}$ is only marginally smaller than $\mathcal{S}$. Unfortunately, near the imaginary axis

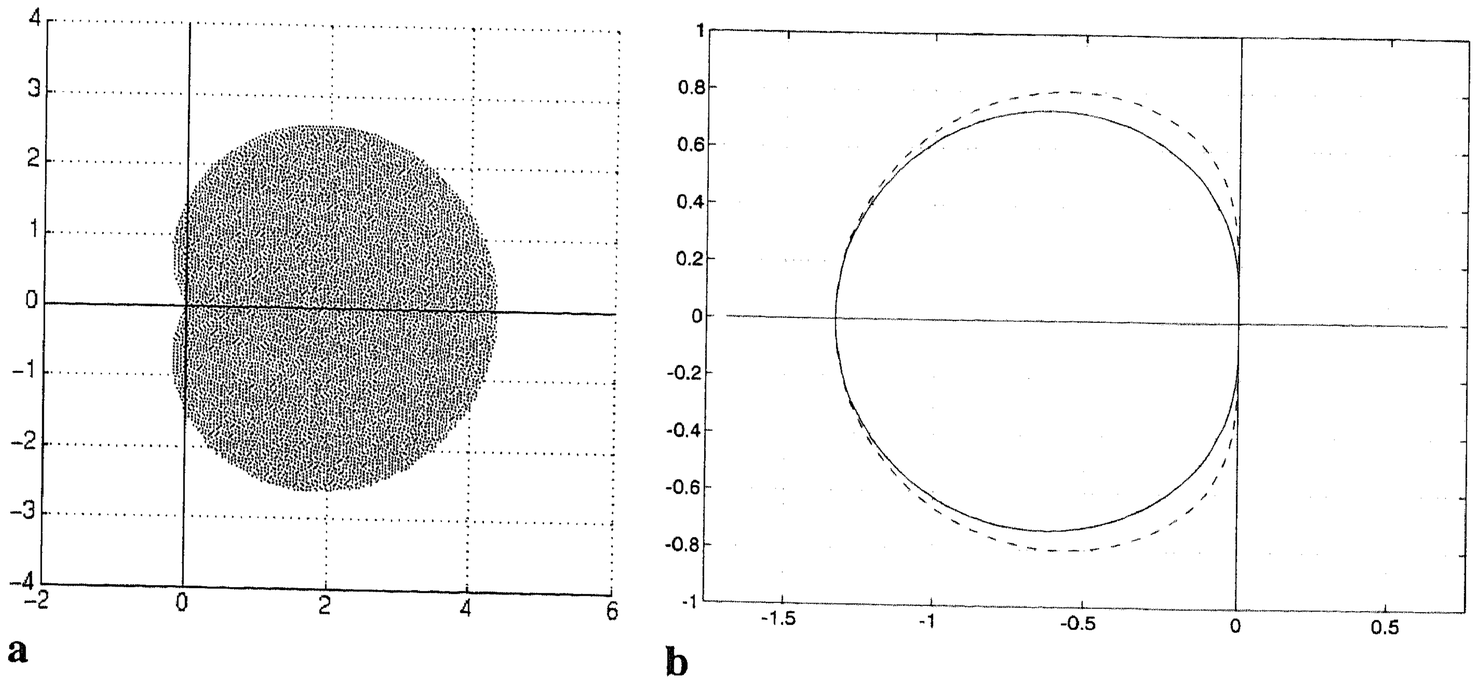


Fig. 10 a, b. Stability regions for IMEX-BDF2. **a** The exterior of the shaded region is the stability region for z_I, assuming arbitrary $z_A \in \mathcal{S}$. **b** The explicit stability region $\mathcal{S}$ (interior of dashed boundary) and the explicit stability region $\mathcal{D}$ (interior of solid boundary)

its boundary moves away from the boundary of $\mathcal{S}$. The consequence is a smaller CFL number for the third-order upwind discretization. The CFL number 0.46 (see Sect. 4.4.3) valid for region $\mathcal{S}$ reduces to 0.23 for region $\mathcal{D}$. Hence with regard to linear stability, the situation depicted in Fig. 10a is to be preferred.

5.3.3 Tridiagonal Gauss–Seidel iteration

A successful application of method (75) to a prototype of a regional 3-D air pollution model is demonstrated in ref. [104]. This model is based on an analytically given horizontal wind field, an analytically prescribed vertical-diffusion term, and a realistic tropospheric gas-phase chemistry model. The latter is the so-called EMEP chemistry model which takes $m = 66$ species into account.

We [104] proposed a special iterative solution technique for the still implicitly coupled chemistry and vertical-diffusion computation. This special technique is meant to replace the standard technique, based on modified Newton iteration and use of a linear block-tridiagonal solver. Recall that $F_D(w)+F_R(w)$ is uncoupled over the horizontal grid, so that the implicit coupling is only one-dimensional. However, despite the 1-D nature, the standard technique is costly when there are very many species.

The special iterative solution technique is an extension of the Gauss–Seidel iteration from Sect. 3.2.3, towards 1-D diffusion-chemistry systems. The Gauss–Seidel iteration is extended such that only tridiagonal linear systems need to be solved. For gas-phase atmospheric chemistry models, this technique works very efficiently and much faster than the standard technique based on using a band solver. If, however, the Gauss–Seidel technique is not applicable and one uses the standard technique, the IMEX scheme is most likely not competitive with Strang or source splitting,

simply because with splitting the vertical diffusion and chemistry computation can be decoupled.

More numerical details on solving 1-D atmospheric diffusion-chemistry systems for a large number of species can be found in refs. [1], [105], and [111]. In ref. [105] the tridiagonal Gauss–Seidel technique is further investigated. In refs. [1] and [111] the technique of approximate matrix factorization is proposed. We discuss the latter technique below for the Rosenbrock method ROS2.

5.4 Approximate matrix factorization

The IMEX technique of the previous section might be viewed upon as splitting within a method. In this section we illustrate the idea of approximate matrix factorization, which might be seen as splitting at the level of the numerical algebra.

5.4.1 The factorized ROS2 method

Following ref. [106], we again consider the Rosenbrock method ROS2 (see (19)). Applied to the semidiscrete system (65) this method reads

$$w^{s+1} = w^s + \tfrac{3}{2}k_1 + \tfrac{1}{2}k_2 \,, \tag{77}$$

$$(I - \gamma \Delta t\, A)k_1 = \Delta t\, F(w^s) \,,$$

$$(I - \gamma \Delta t\, A)k_2 = \Delta t\, F(w^s + k_1) - 2k_1 \,.$$

We use a slightly different formulation here that avoids the matrix-vector multiplication in (19). The matrix A is now supposed to be the huge Jacobian matrix $F'(w^s)$,

$$A = F'_{\mathrm{A}}(w^s) + F'_{\mathrm{D}}(w^s) + F'_{\mathrm{R}}(w^s) \,.$$

Solving the linear systems directly for this A is not advocated, as it would make ROS2 certainly much slower than any of the other 3-D schemes discussed before. Instead, we define A such that $I - \gamma \Delta t\, A$ is approximately factorized into

$$I - \gamma \Delta t\, A = (I - \gamma \Delta t\, F'_{\mathrm{D}}(w^s))(I - \gamma \Delta t\, F'_{\mathrm{R}}(w^s)) \,. \tag{78}$$

With this factorization, the computation of the two stage vectors k_1 and k_2 amounts to a normal F-evaluation and two sequential linear system solutions, one for the vertical diffusion and one for the chemistry. Owing to the box-model structure of F'_{R} and the tridiagonal structure of F'_{D}, these linear systems solutions can be carried out efficiently. It should be emphasized that the original ROS2 method is second-order consistent for any matrix A. Hence with factorization we still have second-order consistenty.

In the stiff ODE literature, methods like ROS2 allowing an arbitrary A for the same order of consistency are sometimes called W-methods [35]. Of interest for our application is that the normal F-evaluation avoids a genuine decoupling as in operator splitting. Since the advection Jacobian $F'_{\mathrm{A}}(w^s)$ is neglected in the factor-

ization, we again assume that advection can be dealt with explicitly. This makes sense here, since for a zero Jacobian matrix ROS2 appears to be identical to the explicit RK2 method studied in Sect. 4.4.2. As shown in Sect. 4.4.2, RK2 has good stability properties for the advection part of the problem. On the other hand, the two linear system solutions, involving the diffusion Jacobian $F'_{\mathrm{D}}(w^s)$ and the chemistry Jacobian $F'_{\mathrm{R}}(w^s)$, are indispensable for stability. Similar as for the IMEX method, the L-stability property of (77) is lost by the approximate factorization.

The technique of approximate matrix factorization is not new. As far as we know, in the numerical solution of PDEs the idea has been introduced by Beam and Warming [9, 108]. Using different time integration methods, for transport problems it is also proposed in refs. [1] and [38].

5.4.2 Stability of ROS2 with factorization

For examining stability, we again consider the standard scalar test model encountered in Fourier–von Neumann analysis, now in the form

$$\dot{w} = \lambda_{\mathrm{A}} w + \lambda_{\mathrm{D}} w + \lambda_{\mathrm{R}} w \,. \tag{79}$$

As previously, λ_{A} denotes an advection eigenvalue, while λ_{D} and λ_{R} are associated with diffusion and chemistry, respectively. Denote $z_{\mathrm{A}} = \Delta t \lambda_{\mathrm{A}}$, etc., and $z = z_{\mathrm{A}} + z_{\mathrm{D}} + z_{\mathrm{R}}$. Application of ROS2 with factorization to the test model (79) yields the simple one-step recurrence

$$w^{s+1} = R(z_{\mathrm{A}}, z_{\mathrm{D}}, z_{\mathrm{R}}) w^s \,,$$

defining the stability function

$$R(z_{\mathrm{A}}, z_{\mathrm{D}}, z_{\mathrm{R}}) = 1 + \frac{2z}{(1-\gamma z_{\mathrm{D}})(1-\gamma z_{\mathrm{R}})} + \frac{\frac{1}{2}z^2 - z}{(1-\gamma z_{\mathrm{D}})^2 (1-\gamma z_{\mathrm{R}})^2} \,. \tag{80}$$

This stability function replaces the L-stable stability function

$$R(z) = \frac{1 + (1-2\gamma)z}{(1-\gamma z)^2}$$

of the original ROS2. Putting $z_{\mathrm{D}} = z_{\mathrm{R}} = 0$ in (80) yields the stability polynomial

$$P(z_{\mathrm{A}}) = 1 + z_{\mathrm{A}} + \tfrac{1}{2} z_{\mathrm{A}}^2$$

of the explicit trapezoidal rule RK2.

We are interested in stability, i.e., $|R(z_{\mathrm{A}}, z_{\mathrm{D}}, z_{\mathrm{R}})| \le 1$, whenever $z_{\mathrm{D}} < 0$ is real and $|P(z_{\mathrm{A}})| \le 1$. This leads to a stability region for z_{R} which we have determined numerically (Fig. 11). The solid line in the left plot is the z_{R}-stability boundary near the imaginary axis, in the neighborhood of the origin. For z_{R} large and purely imaginary, this boundary tends to the imaginary axis. Figure 11 reveals $A(\alpha)$-stability for z_{R}: z_{R} must lie in the wedge $\{ z_{\mathrm{R}} \in \mathbf{C} \mid |\arg(-z_{\mathrm{R}})| < \alpha \}$ with $\alpha = 11°$, approximately. (This angle $\alpha \approx 11°$ is smaller than reported in ref. [106]. The computations

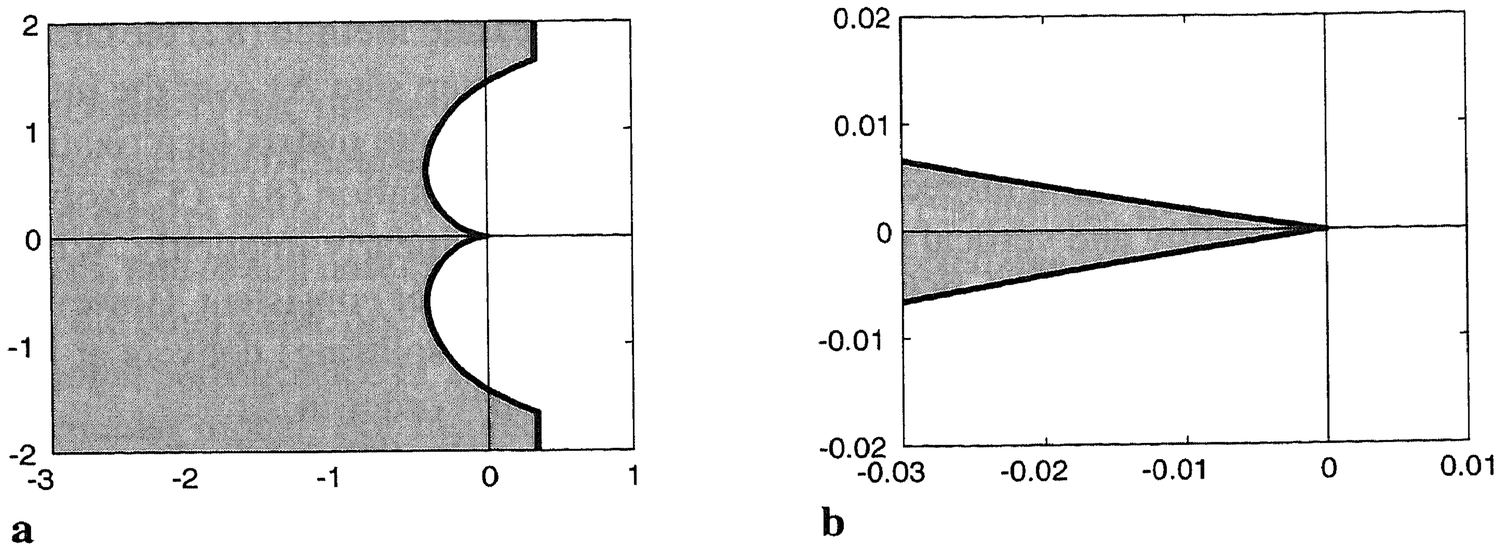


Fig. 11 a, b. The z_R-stability boundary for the factorized stability function. **b** Magnified subregion near the origin

in that paper were not sufficiently accurate.) As observed in Sect. 5.3.2, this angle probably is sufficiently large since eigenvalues with a large imaginary part do not seem to occur in atmospheric-chemistry models. We have imposed $z_D < 0$ and $|P(z_A)| \leq 1$. With a weaker condition on z_A a larger angle will be found.

In conclusion, with respect to linear stability the critical step size of the factorized ROS method (77)–(78) is equal to that of RK2 for the advection computation. Finally we note that the idea of approximate factorization in our 2-stage Rosenbrock method has also been brought forward by Sandu [80].

5.4.3 Source-splitting ROS2 method

We next present a ROS2 variant that combines source splitting and approximate factorization. So we again consider system (71). This system can be numerically integrated by any appropriate stiff ODE solver, once a good choice for the transport source operator $\tilde{F}_T$ has been made. The ROS2 variant is based on the combination

$$\begin{aligned} \tilde{w}^{s+1} &= w^s + \tfrac{3}{2}k_1 + \tfrac{1}{2}k_2 \,, \\ (I - \gamma \Delta t\, A)k_1 &= \Delta t\, F_T(w^s) \,, \\ (I - \gamma \Delta t\, A)k_2 &= \Delta t\, F_T(w^s + k_1) - 2k_1 \,, \\ A &= F'_D(w^s) \,, \end{aligned} \tag{81}$$

and

$$\begin{aligned} w^{s+1} &= w^s + \tfrac{3}{2}k_1 + \tfrac{1}{2}k_2 \,, \\ (I - \gamma \Delta t\, A)k_1 &= \Delta t\, F_R(w^s) + \Delta t\, \tilde{F}_T(w^s) \,, \\ (I - \gamma \Delta t\, A)k_2 &= \Delta t\, F_R(w^s + k_1) + \Delta t\, \tilde{F}_T(w^s) - 2k_1 \,, \\ A &= F'_R(w^s) \,, \end{aligned} \tag{82}$$

where $\tilde{F}_T(w^s)$ is defined by (72).

Method (81) solves the transport problem $\dot{w} = F_T(w)$ and method (82) the chemistry problem $\dot{w} = F_R(w) + \tilde{F}_T(w(t_s))$, using the same step size Δt over the interval $[t_s, t_{s+1}]$. Method (81) is a special case of the approximate matrix factorization scheme (77)–(78). A little inspection shows that the combination (81)–(82) solves advection explicitly and vertical diffusion and chemistry linearly implicitly. When considered on their own, both ROS2 schemes are second-order consistent. However, adding the constant source in the second scheme (source splitting) delivers at the end of each combined step an approximation w^{s+1} of first order in Δt.

Verwer et al. [106] examined the linear stability for (81)–(82) by the linear test model (79). As yet the results are inconclusive. The method can be shown to be unconditionally stable for real negative λ_D and real negative λ_R, as long as the underlying explicit trapezoidal rule is stable. Hence in this sense it seems less generally applicable than the other 3-D methods. However, numerical results reported in ref. [106] deny this and give rise to the conjecture that source splitting is a valuable technique, worthwhile to be further examined for atmospheric transport-chemistry problems. See also the numerical example of Sect. 5.2.4.

5.4.4 Numerical results

Extensive 3-D numerical test results for ROS2, implemented with Strang splitting, approximate factorization, and source splitting, can be found in ref. [106]. These tests are based on a spherical, global prototype model (63), extended with vertical diffusion and real tropospheric gas-phase chemistry (17 species). As velocity field the solid body rotation from ref. [110] is used. The spatial advection scheme is based on the flux-limited, third-order upwind discretization on a reduced grid. The overall conclusion of the tests was that relatively slowly varying species, like O_3, are computed equally accurate with the three implementations. Rapidly varying species, such as HO_2NO_2, are more accurately solved with factorization and source splitting. The difference in accuracy depends on the vertical-diffusion coefficient. A strong vertical diffusion may lead to notable splitting errors with Strang splitting, due to uncoupling diffusion and chemistry.

Tests with realistic models, containing real meteo-data, real parameterized diffusion, orography, etc., are needed to draw more definite conclusions on the occurrence of splitting errors. One such test is presented in ref. [14]. The findings there are not very different from those in ref. [106]. Approximate matrix factorization and source-splitting techniques seem promising from the numerical point of view and warrant further investigation. However, in order to replace the highly flexible standard splitting technique in large air pollution models, they should offer a truly notable error decrease for the same amount of CPU time or, equivalently, a notable decrease in CPU time for the same level of accuracy.

6 High-performance computing

Since the 1970s, the new generations of computers have made computations much faster and much cheaper. This development is of clear interest to air quality modeling, as it creates obvious possibilities to increase grid resolutions, to cover longer time spans, and to add new chemical species so far neglected due to computer constraints. On the other hand, one should always be aware of the fact that increasing the computational performance of a given model, not necessarily leads to better results for this particular model. For instance, subgrid-scale parametrizations, as commonly used in air quality models, are often valid for a certain range of grid sizes. Consequently, making the grid finer and finer, just because computers are nowadays faster and their memories larger, does not necessarily mean that computed solutions actually become better. The opposite can be true because the parametrizations may no longer be valid. Further, without new input data, e.g., describing velocities and emissions more accurately, a finer computational grid does not make much sense either. However, in spite of these warnings, there is no doubt that the current high-performance computers, including high-performance desktop PCs and workstations, can be used to great advantage in air quality modeling.

6.1 Hardware and compiler limitations

When designing large-scale software that will be used for a considerable period of time, it is important to take into account the trends in the development of computer architectures. In the last decade, the rate of improvement in the performance of processors has been much larger than in the performance of memory and I/O systems. Or translated to programming: computational times are getting faster than data access times.

Apparently, memory providing faster data access is more costly to build. Vector/parallel shared-memory (VPSM) systems, like a Cray C90 or a NEC SX4, are so expensive because the access time to the main memory is short enough to feed the processors with sufficient data. Cheaper systems (at least per processor) depend on one or more levels of cache to provide the processors with data. Shared-memory systems also have the advantage that no interconnecting network is needed to use in one processor the data generated by another. On the other hand, the number of processors in such architectures is necessarily limited. In massively parallel processing (MPP) systems, each processor has its own memory and, because the processing elements should be cheap, one or more (small) caches. The main difference between an MPP architecture and the poor man's supercomputer – a farm of PCs or workstations – is the bandwidth of the interconnecting network, and often also the I/O capacity.

Compilers for VPSM systems have matured in the last decade, and it is not uncommon to reach half the peak performance of the system by simply compiling a code. On distributed-memory systems, a mere 10% of the peak performance is a very good score. In addition, this often requires a few days of hand optimization.

The reason for this is not so much the communication time as the memory hierarchy control. Different versions of architectures often use different cache policies, which implies that a well-tailored compiler can perform very badly on the next generation of the same brand. Another problem is the coherency of the data in the cache.

A second bottleneck in the use of distributed systems is I/O. In air pollution modeling, the necessary input data is often stored in huge (sequential) files. This means that either all the data has to be distributed in advance or the currently needed data has to be distributed in a master–slave model during the run. The same holds in principle for the output data, since the currently used visualization packages are also based on single files.

6.2 The virtue of parallelism

Still the buzzword in large-scale numerical modeling is massively parallel computing. Processors are nowadays cheap and powerful, and nothing seems more logical than combining the CPU power of many processing elements into one computer engine with an impressively high peak performance. The idea of exploiting parallelism is already very old. Famous early examples are the "parallel human computers" envisioned by Richardson in very early weather prediction research and used by Turing for the "breaking of the code".

In modern times, the first approach was hand optimization of the model tailored to a specific computer, something which is similar to programming the ENIAC: one has to know everything about the machine and the code to squeeze everything out of it. In this approach it is not really needed to have much knowledge of the underlying application field. Although it is still successfully followed (see, e.g., [25, 26]), it has two disadvantages. First of all, it is very difficult, if not impossible, to prevent divergence of the various versions of the model. Secondly, since computers are nowadays changing faster than the average lifetime of an environmental model, it takes relatively much time to optimize the model by hand for every new computer. It is most likely that one does not want to be convicted to just one computer.

Dependent on the scenario, simulation runs can range from small to large. Simulating short-term episodes, with a simple chemistry model on a coarse grid, can nowadays easily be done on workstations and PCs. Simulations with a complex chemistry model, on a fine grid, will readily give rise to memory restrictions. Long-term simulations, say, over a period of years, require a huge amount of time steps. Extreme CPU power is then needed to obtain feasible turnaround times. In such cases one truly needs modern supercomputers, like VPSM architectures and massively parallel distributed-memory architectures, or clusters of fast workstations or PCs.

The number of parallel processors that can be efficiently used is problem dependent. But in general we certainly do not think of many thousands of processors. Instead, even in the far future, a number of a few hundreds up to at most a thousand seems more realistic. Bulk MPP, with really many thousands of processors, may have its use for nondeterministic search or other truly embarrassingly parallel jobs. We

believe that MPP, with so many processors, cannot be very practical for air quality models (see also [5, 23, 112]). First of all, the underlying system of PDEs is coupled in time, in space, and over the species. This means that independent of the distribution of the data over the processors, communication will always be necessary. For a very large number of processors, communication certainly cannot be neglected. In this connection, one should also realize that off-line air pollution models are I/O bound. Even on-line models, where the meteorological variables are simultaneously computed, are I/O bound. In either case, the necessary input from deposition and emission databases, combined with the desire to visualize or to steer computations on the fly, most likely impedes bulk MPP from becoming very efficient.

6.3 The need for portable code

Development of sophisticated simulation software requires a high investment in terms of human effort. Hence it makes sense to develop software applicable for different scenarios on different computer systems. This leads to the requirement of portability in the development stage. Loosely speaking, the idea of portability is to develop one code for the model which is robust, flexible, and efficient.

Robustness means that both the algorithms and the implementation should be on the safe side. Flexibility means that it should be easy to incorporate changes, e.g., in the chemistry model, or to add (subgrid) processes like cumulus convection, or to change the numerical algorithms. With efficiency we mean that it is possible to run the code on a wide variety of computer platforms with a reasonable performance. In developing and implementing such a code, people tend to follow two different roads: the computer science approach and the scientific computing approach.

The computer science approach is based on the idea that everyone has his own specialism. You have model builders, mathematicians, numerical analysts, and programmers. The specific task of the programmers is to optimize programs and to arrange optimal use of the variety in available computer architectures. For this purpose, computer science can offer

- automatic optimization of code (dusty decks) (e.g., [8, 28, 99]) and
- application languages with building blocks easily optimized for different architectures (e.g., [18, 27, 30]).

Both ideas are nice, but both have disadvantages. In the first option it is almost always necessary to let the automatic optimization be followed by hand optimization. This makes divergence of the codes likely and requires additional human effort. Moreover, since "old" numerical methods and an "old" model are optimized, it will most likely not be very optimal. Especially if it is also written in "spaghetti code", which is often the case when programs are developed over the years by different people. As regards the second option, we do not know of any finished project and have our doubts whether one day such application languages will be mature enough.

With our current knowledge, we thus believe that the computational science ap-

proach is more promising. Here the main idea is to develop the computational model anew, such that

- it is state-of-the-art with respect to the numerics and
- easily allows optimization on different computers.

This involves programming in an existing high-level language, using data parallelism, automatic vectorization and parallelization, and, possibly, templates and libraries (see [6, 7, 66]). Following this approach, the first and main question to be addressed is: how are air quality models solved and how are we to design a code that allows parallelism, especially on loosely coupled processors. Since air pollution modeling is based on many physical processes and input data, testing numerical algorithms and implementational ideas in a full-scale model is far from easy. As an aid to this, special prototypes and benchmark codes are helpful.

6.4 A 3-D benchmark

By a benchmark we mean an implementation of a well-documented prototype which is based on appropriate numerical algorithms and for which the suitability of different computer architectures can be compared.

We developed a benchmark based on a 3-D prototype on the sphere. This prototype is a 3-D regional model, containing those processes that are important from the numerical and computational point of view. The PDE system is of type (66). It thus contains horizontal advection, vertical diffusion, and, most importantly, a real ozone chemistry model. The chemistry model is the state-of-the-art EMEP MSC-W model (140 reactions between 66 species [83, 84]). In one part of the domain we prescribe low rural emissions and in another part high urban emissions. This causes concentrations with large spatial gradients. Large gradients in time arise due to photolysis reactions (day-night rhythm).

We used this prototype quite extensively to study various numerical algorithms [104]. A standard grid on the sphere is used, being uniform in the longitude and latitude direction and nonequidistant in the vertical direction. The numerical method is of MOL type. The advection operator is discretized in space using the flux-limited, third-order upwind scheme (see Sect. 4). The vertical-diffusion operator is discretized by means of a standard second-order central difference method. To integrate the resulting huge and stiff ODE system, the IMEX-BDF2 method is used as described in Sect. 5.3. Consequently, advection is handled explicitly in the time integration, and the chemistry implicitly and directly coupled with the vertical diffusion. To solve the resulting nonlinear systems, the tridiagonal block Gauss–Seidel iteration mentioned in Sect. 5.3.3 is applied. In our tests a fixed number of iterations is used.

6.5 *Implementational paradigms*

Implementational paradigms for the benchmark are discussed in refs. [10], [47], and [104]. For a precise description of the prototype and its benchmark implementation, see also http://www.cwi.nl/~gollum/LOTOS/. From these reports we cite the following main conclusions and recommendations.

The aim is to have one code which can be efficiently used on a variety of computer platforms. The most demanding architecture for such a code is a distributed platform, where the communication bandwidth is small compared to the CPU performance. This is normally the case for a cluster of workstations. On such parallel distributed-memory systems, the one thing that is important when developing a code, is the coupling which exists between the data that is distributed over the processing elements (PEs). In Eq. (66), the advection and diffusion parts give rise to a coupling in space and the chemical reaction term to a coupling across the species. It is customary to compute the chemistry part implicitly, possibly coupled with vertical diffusion. The advection part is mostly always integrated explicitly in time. The implicit integration of the chemistry is the most time-consuming process.

The physical data distribution across the PEs should be such that the most expensive part of the computation will be done on local data. Redistribution of the data over the PEs, which is an option on a distributed-memory architecture with a very fast interconnecting network, such as the Cray T3E, is not an option on a cluster of workstations coupled in an ethernet or ATM star-network. So the four-dimensional concentration vectors – 3 space dimensions and the species – should be distributed, such that the concentration values and the vertical dimension for each grid point in the horizontal domain reside on one processor.

The most transparent way to realize this in one code for all platforms is the data parallel programming style. Here the data is distributed using directives and the computations are expressed as much as possible in array syntax constructs, which are automatically distributed over the PEs. This can be done completely portable in HPF, and almost completely portable in Fortran 90. A disadvantage is that such an implementation is not very efficient on loosely coupled processors, and most likely never will be.

A second way of looking at a portable implementation of the model, is the SPMD/domain decomposition approach. Here the physical domain of the global model is decomposed in subdomains that are distributed over the PEs. On each PE, a local model is computed, with in every time step known boundary conditions between the subdomains if explicit time integration is used. Even on a distributed-memory architecture with a dedicated and very fast network, this is the best option [10]. Communication of the internal boundaries is done in an enveloping program, which has no other task than to take care of the exchange of the internal boundary conditions and to activate the different computational modules. The computations for the regional (sub)model are expressed as much as possible in array syntax constructs, to facilitate automatic vectorization and possibly also parallelization on shared-memory or virtually shared-(physically distributed-)memory architectures.

6.6 *Benchmark results*

We compared the performance of our benchmark on four different architectures: a vector/parallel shared-memory architecture (a Cray C90 with 12 processors), a parallel shared-memory architecture (SGI Origin 2000), a massively parallel distributed-memory system (a Cray T3E), and a cluster of SGI O2 workstations coupled in a star-shaped ATM network. To value the difference in performance, we give in Table 1 some technical information of these platforms: the clock frequency, the amount of primary and secondary cache, the available memory, and for the distributed platforms the speed of the interconnecting network. Figure 12 shows a typical performance of the benchmark code on the various computer platforms. Figure 12 presents on

Table 1. Hardware specifications: clock frequency, size of primary and secondary cache, memory (for distributed memory systems per processor), and bandwidth of interconnecting network

Architecture	Clock	Cache	Memory	Remote access
Cray C90	240 MHz		8 GB	
SGI Origin 2000 R10000	195 MHz	32 KB + 4 MB	32 GB	
Cray T3E	300 MHz	8 KB + 96 KB	128 MB	6 × 500 MB/s
SGI O2 R5000	180 MHz	32 KB + 0.5 MB	96 MB	5 MB/s

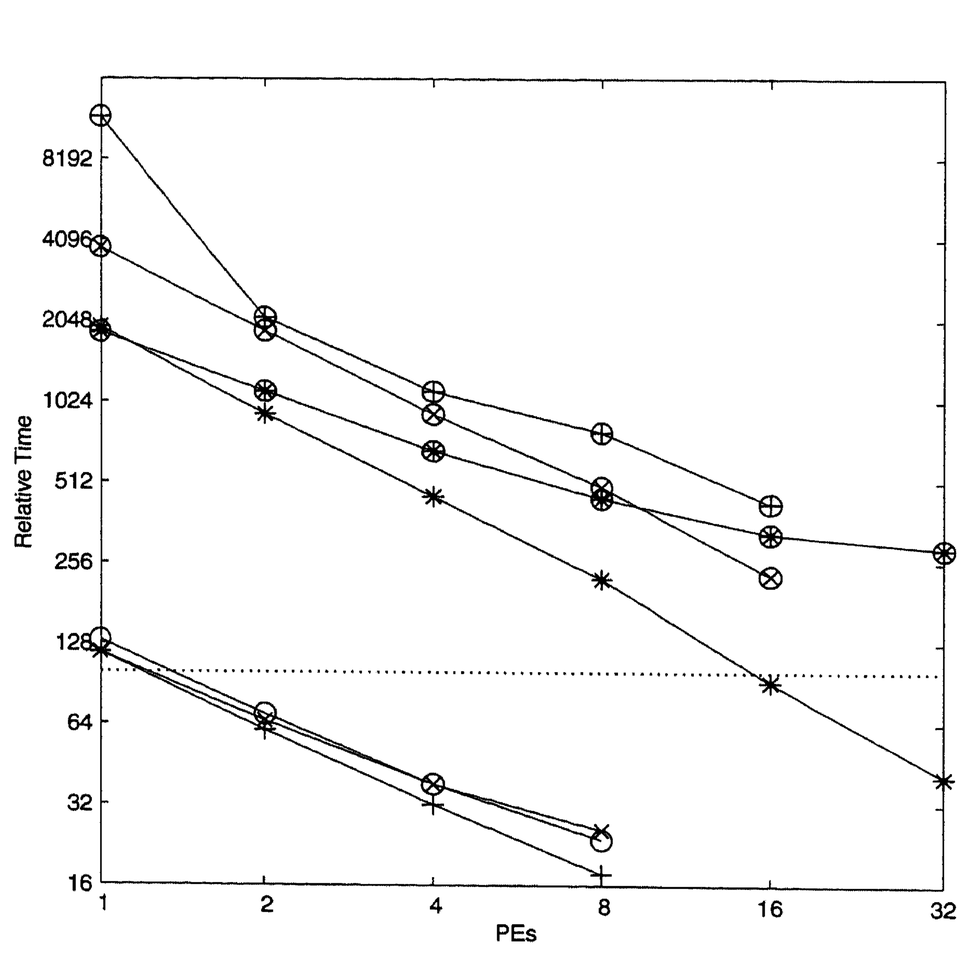


Fig. 12. Benchmark performances: CPU times shown relative to the CPU time needed by the C90, using 1 CPU and compiler option -Zv. The norm for the C90 is 100 (the dotted line), which represents a performance of 500 Mflop. +, C90 (-ZP): Amdahl (atexpert); ×, C90 (-ZP): "dedicated" (atexpert); ∘ C90 (pvm); ∗ T3E (pvm); ⊛SGI Origin 2000; ⊕ SGI R5000 + ATM (pvm): WCT; ⊗ SGI R5000 + ATM (pvm): CPU

a log 2-to-log 2 scale, CPU times versus the number of PEs, taken relative to one CPU of the Cray C90. On the C90, the benchmark reaches 500 Mflop, which is half the peak performance. Investigations reported in ref. [13] point to the following conclusions.

The scalability of the model with the number of PEs is very good, but for a larger number of processors communication time and parallelism overhead will show up. More important, however, is the large gap that exists between the performance on a vector/parallel architecture and the other systems. The gap is not caused by the speed of the PEs but mainly by the memory hierarchy. The picture shows nicely that a key factor in high-performance computing is high-performance memory: a large number of memory banks and separate memory ports for all processors leads to a much better performance than bus-based cache-dominated systems.

We do not claim that this plot shows the ultimate performance for all air quality models on these types of computers. We do believe though, that our 3-D prototype is a good approximation of a full-scale model. The one thing lacking in this prototype are the necessary I/O operations. As input an off-line model needs meteo and emission data. Further, to enable on-the-fly change of scenarios, the computed concentrations need to be visualized. Therefore, for a good performance one also needs sufficient I/O capacity and the possibility to use it in a portable way. To test I/O performance, we performed some experiments on the Cray T3E (see [13]). However, the results are as yet not decisive. A preliminary conclusion is that the I/O does not scale, although the Cray T3E has a scalable I/O architecture. On the other hand, for a not too large number of PEs, the I/O times were not significant with asynchronous parallel I/O. At the moment it is not possible to implement this in a portable way, but asynchronous I/O will be available in the next HPF version and perhaps also in Fortran 2000. It is also possible that the implementation of parallel I/O in MPI will give sufficient I/O efficiency.

6.7 Main conclusions and final remarks

The main conclusions that we have drawn from the high-performance computing experiments are the following.

A key factor in high-performance computing is high-performance memory: a large number of memory banks and separate memory ports for all processors leads to a much better performance than bus-based cache-dominated systems. Consequently, for real computer speed for air quality models, the VPSM architectures are by far preferable. On the other hand, if the size of the model leads to memory constraints, the distributed-memory architectures are a good alternative. Both are expensive. Cheaper, and not far from competitive, is a dedicated cluster of workstations.

With respect to the necessary I/O in an air quality model, the use of the (almost portable) asynchronous I/O is currently an efficient choice on all architectures. It results in negligible I/O times on "shared-nothing" architectures, where each processor has its own memory and disk.

To avoid divergence of different implementations of an air quality model aimed

at different computer platforms, it is highly recommendable to have one implementation of the model. The experiments with our benchmark code on various platforms show that this is possible.

In the design of the algorithms, and the implementations thereof, one should bear in mind that the most impeding part of computer architectures on the performance of an air quality model is the memory hierarchy. If the computations do not fit in memory, most of the time is spent on swapping data from disk to memory. Here, and in the dynamic behavior of the cache, one can distinguish two different reasons for data misses: compulsory misses, e.g., because the data has to be obtained from another processor, and capacity or conflict misses. These can be possibly cured by data reordering or clustering.

With respect to the numerical algorithms to be used, we prefer that the time integration methods should allow large step sizes equally chosen over the space grid, also for the stiff chemistry computations. When time steps in the order of a few minutes up to, say, half an hour are possible for the chemistry, the above described SPMD/domain decomposition approach is very efficient and the problem of load balancing (see [25, 26]) is avoided. It also means that there is no real implementational difference between the different time integration approaches described in Sect. 5, including operator splitting, source splitting, IMEX, and approximate matrix factorization.

A further important aspect in the development of the numerical algorithms is to take care that data access should as much as possible fulfill the requirement of spatial and temporal locality. The first makes it possible to efficiently use vector registers and cache lines, while the second is of influence on the reusage of cache.

Acknowledgments

This work was supported by the Dutch Stichting HPCN, with financial support from the Dutch Ministry of Economic Affairs through the HPCN-TASC project "HPCN for Environmental Applications". We also thank Cray Research, Inc., for the grant CRG 97.02 and the National Computing Facilities foundation, who provided use of supercomputer facilities with financial support from the Netherlands Organization for Scientific Research.

References

1. Ahmad, I., Berzins, M.: An algorithm for ODEs from atmospheric dispersion problems. Appl. Numer. Math. 25: 137–150 (1997).
2. Allen, D. J., Douglass, A. R., Rood, R. B., Guthrie, P. D.: Application of a monotonic upstream-biased transport scheme to three-dimensional constituent transport calculations. Mon. Weather Rev. 119: 2456–2464 (1991).
3. Ascher, U. M., Ruuth, S. J., Wetton, B.: Implicit-explicit methods for time-dependent PDEs. SIAM J. Numer. Anal. 32: 797–823 (1997).
4. Ascher, U. M., Ruuth, S. J., Spiteri, R. J.: Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations. Appl. Numer. Math. 25: 151–167 (1997).
5. Baillie, C., Michalakes, J., Skålin, R.: Regional weather modeling on parallel computers. Parallel Comput. 23: 2135–2142 (1997).

6. Balay, S., Gropp, W. D., Curfman McInnes, L., Smith, B. F.: PETSc home page. http://www-fp.mcs.anl.gov/petsc (December 2001).
7. Balay, S., Gropp, W. D., Curfman McInnes, L., Smith, B. F.: Efficient management of parallelism in object oriented numerical software libraries. In: Arge, E., Bruaset, A. M., Langtangen, H. P. (eds.): Modern software tools in scientific computing. Birkhäuser, Basel, pp. 163–202 (1997).
8. Banerjee, P., Chandy, J. A., Gupta, M., Hodges, E. W., Holm, J. G., Lain, A., Palermo, D. J., Ramaswamy, S., Su, E.: The PARADIGM compiler for distributed-memory multicomputers. IEEE Computer 28 (10): 37–47 (1995).
9. Beam, R. M., Warming, R. F.: An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. J. Comput. Phys. 22: 87–110 (1976).
10. Blom, J. G., Kessler, C., Verwer, J. G.: An evaluation of the Cray T3D programming paradigms in atmospheric transport-chemistry models. Report NM-R9604, Centrum voor Wiskunde en Informatica, Amsterdam (1996).
11. Blom, J. G., Roemer, M. G. M.: Description of the 3D LOTOS model, part I: dynamics. Report MAS-N9701, Centrum voor Wiskunde en Informatica, Amsterdam (1997).
12. Blom, J. G., Hundsdorfer, W., Verwer, J. G.: Vectorization aspects of a spherical advection scheme on a reduced grid. Report NM-R9418, Centrum voor Wiskunde en Informatica, Amsterdam (1994).
13. Blom, J. G., Lioen, W. M., Verwer, J. G.: HPCN and air quality modeling. In: Sloot, P., Bubak, M., Hertzberger, B. (eds.): High-performance computing and networking. Springer, Berlin Heidelberg New York Tokyo, pp. 141–150 (Lecture Notes in Computer Science, vol. 1401) (1998).
14. Blom, J. G., Verwer, J. G.: A comparison of integration methods for atmospheric transport-chemistry problems. J. Comp. Appl. Math 126: 381–396 (2000).
15. Bolley, C., Crouzeix, M.: Conservation de la positivité lors de la discrétisation des problèmes d'évolution paraboliques. R. A. I. R. O. Anal. Numer. 12: 237–245 (1978).
16. Boris, J. P., Book, D. L.: Flux corrected transport I: SHASTA, a fluid transport algorithm that works. J. Comput. Phys. 11: 38–69 (1973).
17. Bott, A.: Monotone flux limitation in the area-preserving flux-form advection algorithm. Mon. Weather Rev. 120: 2595–2602 (1992).
18. Chen, M. C., Choo, Y., Li, J.: Crystal: theory and pragmatics of generating efficient parallel code. In: Szymanski, B. K. (ed.): Parallel functional languages and compilers. Association for Computing Machinery, New York, pp. 255–308 (1991).
19. Colella, P., Woodward, P.: The piecewise parabolic method (PPM) for gas-dynamical simulations. J. Comput. Phys. 54: 174–201 (1984).
20. Damian-Iordache, V., Sandu, A.: KPP: a symbolic preprocessor for chemistry kinetics, user's guide. Report, Department of Mathematics, University of Iowa, Iowa City, Iowa (1995).
21. Dekker, K., Verwer, J. G.: Stability of Runge–Kutta methods for stiff non-linear differential equations. North-Holland, Amsterdam (1984).
22. Deuflhard, P., Heroth, J., Maas, U.: Towards dynamic dimension reduction in reactive flow problems. Preprint SC 96-27, Konrad-Zuse-Zentrum für Informationstechnik, Berlin (1996).
23. Drake, J., Foster, I.: Parallel computing in climate and weather modeling. Parallel Comput. 21: 1539–1544 (1995).
24. Easter, R. C.: Two modified versions of Bott's positive-definite numerical advection scheme. Mon. Weather Rev. 121: 297–304 (1993).
25. Elbern, H.: Parallelization and load balancing of a comprehensive atmospheric chemistry transport model. Atmos. Environ. 31: 3561–3574 (1997).
26. Elbern, H.: On the load balancing problem of comprehensive air quality models. Syst. Anal. Modelling Simulation 32: 31–56 (1998).
27. van Engelen, R. A., Wolters, L., Cats, G.: The Ctadel code generation tool for PDE-based scientific applications. In: Proceedings of the Second Annual Conference of the Advanced School for Computing and Imaging (ASCI), Lommel, Belgium, pp. 120–125 (1996).

28. Applied Parallel Research: FORGE90. Applied Parallel Research, Inc., Placerville, Calif.
29. Frank, J., Hundsdorfer, W., Verwer, J. G.: On the stability of implicit-explicit linear multistep methods. Appl. Numer. Math. 25: 193–205 (1997).
30. Giloi, W. K., Schramm, A.: PROMOTER: an application-oriented programming model for massive parallelism. In: Giloi, W. K., Jahniche, S., Shriver, B. D. (eds.): Programming models for massively parallel computers. IEEE Computer Society Press, Los Alamos (1993).
31. Golub, G. H., van Loan, C. F.: Matrix computations, 2nd ed. Johns Hopkins University Press, Baltimore (1990).
32. Graedel, T. E., Crutzen, P. J.: Atmosphere, climate and change. Freeman and Company, New York (1995).
33. Griffiths, D. F., Sanz-Serna, J. M.: On the scope of the method of modified equations. SIAM J. Sci. Stat. Comput. 7: 994–1008 (1986).
34. Gustafsson, B., Kreiss, H-O., Oliger, J.: Time dependent problems and difference methods. Wiley, New York (1995).
35. Hairer, E., Wanner, G.: Solving ordinary differential equations II: stiff and differential-algebraic problems, 2nd ed. Springer, Berlin Heidelberg New York Tokyo (1996).
36. Hertel, O., Berkowicz, R., Christensen, J., Hov, Ø.: Test of two numerical schemes for use in atmospheric transport-chemistry models. Atmos. Environ. 27: 2591–2611 (1993).
37. Hesstvedt, E., Hov, Ø., Isaksen, I.: Quasi-steady-state-approximation in air pollution modelling: comparison of two numerical schemes for oxidant prediction. Int. J. Chem. Kinet. 10: 971–994 (1978).
38. van der Houwen, P. J., Sommeijer, B. P., Kok, J.: The iterative solution of fully implicit discretizations of three-dimensional transport models. Appl. Numer. Math. 25: 243–256 (1997).
39. Hundsdorfer, W.: Unconditional convergence of some Crank-Nicolson LOD methods for initial-boundary value problems. Math. Comput. 58: 35-53 (1992).
40. Hundsdorfer, W., Trompert, R. A.: Method of lines and direct discretization: a comparison for linear advection. Appl. Numer. Math. 13: 469–490 (1994).
41. Hundsdorfer, W., Koren, B., van Loon, M., Verwer, J. G.: A positive finite-difference advection scheme. J. Comput. Phys. 117: 35–46 (1995).
42. Hundsdorfer, W., Spee, E. J.: An efficient horizontal advection scheme for modelling of global transport of constituents. Mon. Weather Rev. 123: 3554-3564 (1995).
43. Hundsdorfer, W.: Partially implicit BDF2 blends for convection dominated flows. SIAM J. Numer. Anal. 38: 1763–1783 (2001).
44. Jacobson, M. Z., Turco, R. P.: SMVGEAR: a sparse-matrix, vectorized Gear code for atmospheric models. Atmos. Environ. 28: 273–284 (1994).
45. Jay, L. O., Sandu, A., Potra, F. A., Carmichael, G. R.: Improved quasi-steady-state-approximation methods for atmospheric chemistry integration. SIAM J. Sci. Comput. 18: 182–202 (1997).
46. Kessler, C.: Entwicklung eines effizienten Lösungsverfahrens zur modellmäßigen Beschreibung der Ausbreitung und chemischen Umwandlung reaktiver Luftschadstoffe. Shaker, Aachen (1995).
47. Kessler, C., Blom, J. G., Verwer, J. G.: Porting a 3D model for the transport of reactive air pollutants to the parallel machine T3D. Report NM-R9515, Centrum voor Wiskunde en Informatica, Amsterdam (1995).
48. Kessler, C., Griesel, A., Verwer, J. G.: A Rosenbrock solver in chemistry-transport modelling: what about its speed in 3D? In: Proceedings EUROTRAC-2 Symposium, Garmisch-Partenkirchen, March 23–27, 1998 (1998).
49. Lanser, D., Verwer, J. G.: Analysis of operator splitting for advection-diffusion-reaction problems from air pollution modelling. J. Comput. Appl. Math. 111: 201–216 (1999).
50. Knoth, O., Wolke, R.: A comparison of fast chemical kinetic solvers in a simple vertical diffusion model. In: Proceedings of the 20th International Meeting on Air Pollution Modelling and Its Applications, Valencia, Spain (1993).

51. Knoth, O., Wolke, R.: Numerical methods for the solution of large kinetic systems. Appl. Numer. Math. 18: 211–221 (1995).
52. Knoth, O., Wolke, R.: Implicit-explicit Runge–Kutta methods for computing atmospheric reactive flows. Appl. Numer. Math. 28: 327-341 (1998).
53. Knoth, O., Wolke, R.: An explicit-implicit numerical approach for atmospheric chemistry-transport modelling. Atmos. Environ. 32: 1785–1797 (1998).
54. Koren, B.: A robust upwind discretization method for advection, diffusion and source terms. In: Vreugdenhil, C. B., Koren, B. (eds.): Numerical methods for advection-diffusion problems. Vieweg, Braunschweig, pp. 117–138 (Notes on numerical fluid mechanics, vol. 45) (1993).
55. van Leer, B.: Towards the ultimate conservative difference scheme IV: a new approach to numerical convection. J. Comput. Phys. 23: 276–299 (1977).
56. van Leer, B.: Upwind-difference methods for aerodynamic problems governed by the Euler equations. In: Engquist, B. E., Osher, S., Sommerville, R. C. J. (eds.): Large-scale computations in fluid mechanics. American Mathematical Society, Providence, R. I., pp. 327–336 (1985).
57. Leonard, B. P.: Elliptic systems: finite difference method IV. In: Minkowycz, W. J., Sparrow, E. M., Schneider, G. E., Pletcher, R. H. (eds.): Handbook of numerical heat transfer. Wiley, New York (1988).
58. Leonard, B. P., Lock, A. P., MacVean, M. K.: Conservative explicit unrestricted-time-step multidimensional constancy-preserving advection schemes. Mon. Weather Rev. 124: 2588–2606 (1996).
59. Leonard, B. P.: Simple high accuracy resolution program for convective modeling of discontinuities. Int. J. Numer. Methods Fluids 8: 1291–1318 (1988).
60. LeVeque, R. J.: Numerical methods for conservation laws. Birkhäuser, Basel (1992).
61. LeVeque, R. J.: Large time step shock-capturing techniques for scalar conservation laws. SIAM J. Numer. Anal. 19: 1091–1109 (1982).
62. Lin, S. J., Rood, R. B.: Multidimensional flux-form semi-Lagrangian transport scheme. Mon. Weather Rev. 124: 2046–2070 (1996).
63. van Loon, M.: Numerical methods in smog prediction. PhD thesis, University of Amsterdam, Amsterdam, the Netherlands (1996).
64. Maas, U., Pope, S. B.: Simplifying chemical kinetics: intrinsic low dimensional manifolds in composition space. Combust. Flame 88: 239–264 (1992).
65. McRae, G. J., Goodin, W. R., Seinfeld, J. H.: Numerical solution of the atmospheric diffusion equation for chemically reacting flows. J. Comput. Phys. 45: 1–42 (1982).
66. Michalakes, J.: RSL: a parallel runtime system library for regional atmospheric models. In: Baden, S., Chrisochoides, N., Gannon, D., Norman, M. (eds.): Structured adaptive mesh refinement grid methods. Springer, Berlin Heidelberg New York Tokyo, pp. 59–74 (The IMA volumes in mathematics and its applications, vol. 117) (2000).
67. Peters, L. K., Berkowitz, C. M., Carmichael, G. R., Easter, R. C., Fairweather, G., Ghan, S. J., Hales, J. M., Leung, L. R., Pennell, W. R., Potra, F. A., Saylor, R. D., Tsang, T. T.: The current state and future direction of Eulerian models in simulating the tropospheric chemistry and transport of trace species: a review. Atmos. Environ. 29: 189–222 (1995).
68. Petersen, A. C., Spee, E. J., van Dop, H., Hundsdorfer, W.: An evaluation and intercomparison of four new advection schemes for use in global chemistry models. J. Geophys. Res. 103: 19253-19269 (1998).
69. Prather, M. J.: Numerical advection by conservation of second-order moments. J. Geophys. Res. 91: 6671–6681 (1987).
70. Rasch, P. J.: Conservative shape-preserving two-dimensional transport on a spherical reduced grid. Mon. Weather. Rev. 122: 1337–1350 (1994).
71. Rentrop, P., Steinebach, G.: Model and numerical techniques for the alarm system of the river Rhine. Surv. Math. Ind. 6: 245–265 (1997).

72. Richtmyer, R. D., Morton, K. W.: Difference methods for initial-value problems, 2nd ed. Interscience Publishers, New York (1967).
73. Roe, P. L.: Some contributions to the modeling of discontinuous flows. Lect. Notes Appl. Math. 22: 163–193 (1985).
74. Rood, R. B.: Numerical advection algorithms and their role in atmospheric transport and chemistry models. Rev. Geophys. 25: 71–100 (1987).
75. Rosenbrock, H. H.: Some general implicit processes for the numerical solution of differential equations. Comput. J. 5: 329–330 (1963).
76. Russell, G. L., Lerner, J. A.: A new finite-differencing scheme for the tracer transport equation. J. Appl. Meteorol. 20: 1483–1498 (1981).
77. Sandu, A., Potra, F. A., Carmichael, G. R., Damian, V.: Efficient implementation of fully implicit methods for atmospheric chemical kinetics. J. Comput. Phys. 129: 101–110 (1996).
78. Sandu, A., Verwer, J. G., van Loon, M., Carmichael, G. R., Potra, F. A., Dabdub, D., Seinfeld, J. H.: Benchmarking stiff ODE solvers for atmospheric chemistry problems I: implicit versus explicit. Atmos. Environ. 31: 3151–3166 (1997).
79. Sandu, A., Verwer, J. G., Blom, J. G., Spee, E. J., Carmichael, G. R.: Benchmarking stiff ODE solvers for atmospheric chemistry problems II: Rosenbrock solvers. Atmos. Environ. 31: 3459–3472 (1997).
80. Sandu, A.: Numerical aspects of air quality modelling. PhD thesis, University of Iowa, Iowa City, Iowa (1997).
81. Sanz-Serna, J. M., Verwer, J. G., Hundsdorfer, W. H.: Convergence and order reduction of Runge–Kutta schemes applied to evolutionary partial differential equations. Numer. Math. 50: 405–418 (1987).
82. Shu, C. W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. 77: 439–471 (1988).
83. Simpson, D.: Photochemical model calculations over Europe for two extended summer periods: 1985 and 1987: model results and comparisons with observations. Atmos. Environ. 27A: 921–943 (1993).
84. Simpson, D., Andersson-Skold, Y., Jenkin, M. E.: Updating the chemical scheme for the EMEP MSC-W model: current status. EMEP MSC-W Note 2/93, Norwegian Meteorological Intitute, Oslo (1993).
85. Smolarkiewicz, P. K.: The multi-dimensional Crowley advection scheme. Mon. Weather Rev. 110: 1968–1983 (1982).
86. Smolarkiewicz, P. K., Rasch, P. J.: Monotone advection on the sphere: an Eulerian versus semi-Lagrangian approach. J. Atmos. Sci. 48: 793–810 (1991).
87. Spee, E. J., de Zeeuw, P. M., Verwer, J. G., Blom, J. G., Hundsdorfer, W.: A numerical study for global atmospheric transport-chemistry problems. Math. Comput. Simulation 48: 177–204 (1998).
88. Spee, E. J.: Numerical methods in global transport-chemistry models. PhD thesis, University of Amsterdam, Amsterdam, the Netherlands (1998).
89. Sportisse, B., Jaubertie, A., Plion, P.: Reducing mechanism in chemical kinetics for the simulation of reactive transport: an application to air pollution modelling. In: Sportisse, B. (ed.): Proceedings Saint-Venant Symposium, Paris, August 1997 (1997).
90. Sportisse, B.: Reducing chemical kinetics: a mathematical point of view. In: Sportisse, B. (ed.): Proceedings Workshop Numerical Aspects of Reduction in Chemical Kinetics, Sept. 2, 1997, Centre d'Enseignement et de Recherche en Mathematiques, Informatique et Calcul Scientifique, Ecole Nationale des Ponts et Chaussées, Paris (1997).
91. Sportisse, B.: An analysis of operator splitting techniques in the stiff case. Report 98-127, Centre d'Enseignement et de Recherche en Mathematiques, Informatique et Calcul Scientifique, Ecole Nationale des Ponts et Chaussées, Paris (1998).
92. Sportisse, B. (ed.): Proceedings Workshop Numerical Aspects of Reduction in Chemical Kinetics, Sept. 2, 1997, Centre d'Enseignement et de Recherche en Mathematiques, Informatique et Calcul Scientifique, Ecole Nationale des Ponts et Chaussées, Paris (1997).

93. Strang, G.: Trigonometric polynomials and difference methods of maximal accuracy. J. Math. Phys. 41: 147–154 (1962).
94. Strang, G.: On the construction and comparison of difference schemes. SIAM J. Numer. Anal. 5: 506–517 (1968).
95. Sun, P.: A pseudo-non-time-splitting method in air quality modelling. J. Comput. Phys. 127: 152–157 (1996).
96. de Swart, J. J. B., Blom, J. G.: Experiences with sparse matrix solvers in parallel ODE software. Comput. Math. Appl. 31: 43–55 (1996).
97. Sweby, P. K.: High resolution schemes using flux-limiters for hyperbolic conservation laws. SIAM J. Numer. Anal. 21: 995–1011 (1984).
98. Tomlin, A., Berzins, M., Ware, J., Smith, J., Pilling, M. J.: On the use of adaptive gridding methods for modelling chemical transport from multi-scale sources. Atmos. Environ. 31: 2945–2959 (1997).
99. VAST, Veridian Systems, Santa Monica, Calif., http://www.psrv.com/vast.html (December 2001).
100. Verwer, J. G.: Gauss–Seidel iteration for stiff ODEs from chemical kinetics. SIAM J. Sci. Comput. 15: 1243–1250 (1994).
101. Verwer, J. G., van Loon, M.: An evaluation of explicit pseudo-steady-state approximation schemes for stiff ODE systems from chemical kinetics. J. Comput. Phys. 113: 347–352 (1994).
102. Verwer, J. G., Blom, J. G., van Loon, M., Spee, E. J.: A comparison of stiff ODE solvers for atmospheric chemistry problems. Atmos. Environ. 30: 49–58 (1995).
103. Verwer, J. G., Simpson, D.: Explicit methods for stiff ODEs from atmospheric chemistry. Appl. Numer. Math. 18: 413–430 (1995).
104. Verwer, J. G., Blom, J. G., Hundsdorfer, W.: An implicit-explicit approach for atmospheric transport-chemistry problems. Appl. Numer. Math. 20: 191–209 (1996).
105. Verwer, J. G., Blom, J. G.: On the coupled solution of diffusion and chemistry in air pollution models. In: Kreuzer, E., Mahrenholtz, O. (eds.): Applied sciences, especially mechanics, minisymposia contributions: ICIAM/GAMM 95, Hamburg, July 3–7, 1995. Akademie-Verlag, Berlin, pp. 469–472 (Zeitschrift für Angewandte Mathematik und Mechanik, vol. 76, suppl. 4) (1996).
106. Verwer, J. G., Spee, E. J., Blom, J. G., Hundsdorfer, W.: A second order Rosenbrock method applied to photochemical dispersion problems. SIAM J. Sci. Comput. 20: 456–480 (1999).
107. Vreugdenhil, C. B., Koren, B. (eds.): Numerical methods for advection-diffusion problems. Vieweg, Braunschweig (Notes on numerical fluid mechanics, vol. 45) (1993).
108. Warming, R. F., Beam, R. M.: An extension of A-stability to alternating direction methods. BIT 19: 395–417 (1979).
109. Williamson, D. L.: Review of numerical approaches for modelling global transport. In: van Dop, H., Kallos, G. (eds.): Air pollution modelling and its application IX. Plenum, New York, pp. 377–394 (1992).
110. Williamson, D. L., Rasch, P. J.: Two-dimensional semi-Lagrangian transport with shape-preserving interpolation. Mon. Weather Rev. 117: 102–129 (1989).
111. Wolke, R., Knoth, O.: Implicit-explicit Runge–Kutta methods applied to atmospheric chemistry-transport modelling. In: Proceedings APMS '98: International Conference on Air Pollution Modelling and Simulation, Champs-sur-Marne, France, pp. 335–350 (1998).
112. Womble, D. E., Greenberg, D. S.: Parallel I/O: an introduction. Parallel Comput. 23: 403–417 (1997).
113. Young, T. R., Boris, J. P.: A numerical technique for solving stiff ordinary differential equations associated with the chemical kinetics of reactive flow problems. J. Phys. Chem. 81: 2424–2427 (1977).
114. Zalesak, S. T.: Fully multidimensional flux-corrected transport algorithms for fluids. J. Comput. Phys. 31: 335–362 (1979).

115. Zalesak, S. T.: A preliminary comparison of modern shock-capturing schemes: linear advection. In: Vichnevetsky, R., Stapleman, R. S. (eds.): Advances in computer methods for partial differential equations VI. International Association for Mathematics and Computers in Simulation, Department of Computer Science, Rutgers University, New Brunswick, N. J., pp. 15–22 (IMACS series in computational and applied mathematics) (1987).
116. Zlatev, Z.: Computer treatment of large air pollution models. Kluwer, Dordrecht (1995).

Authors' address: J. G. Verwer, W. H. Hundsdorfer, and J. G. Blom, Centrum voor Wiskunde en Informatica, P. O. Box 94079, 1090 GB Amsterdam, the Netherlands.

Verleger: Springer-Verlag KG, Sachsenplatz 4–6, A-1201 Wien. – Herausgeber: Prof. Dipl.-Ing. Dr. H. Engl, Institut für Mathematik, Johannes-Kepler-Universität Linz, A-4040 Linz. – Redaktion: Altenbergerstraße 69, A-4040 Linz. – Hersteller: Satz und Umbruch: HD Ecker TeXtservices, D-53225 Bonn. – Offsetdruck: Novographic, Ing. W. Schmid, Maurer Lange Gasse 64, A-1238 Wien. – Verlagsort: Wien. – Herstellungsort: Wien. – Printed in Austria.

Offenlegung gem. § 25 Abs. 1 bis 3 Mediengesetz:

Unternehmensgegenstand: Verlag von wissenschaftlichen Büchern und Zeitschriften.

An der Springer-Verlag KG ist beteiligt: Springer-Verlag GmbH & Co KG, Heidelberger Platz 3, D-14197 Berlin, als Kommanditist mit 74,04 %. Geschäftsführer: Rudolf Siegle, Sachsenplatz 4–6, A-1201 Wien.