Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

Benchmarking stiff ODE solvers for atmospheric chemistry problems II: Rosenbrock solvers

A. Sandu, J.G. Verwer, J.G. Blom, E.J. Spee and G.R. Carmichael

Department of Numerical Mathematics

NM-R9614 October 31, 1996

Report NM-R9614
ISSN 0169-0388

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

Benchmarking Stiff ODE Solvers for Atmospheric Chemistry Problems II: Rosenbrock Solvers

A. Sandu
Program in Applied Mathematics and Computational Sciences
The University of Iowa, Iowa City, IA 52246, USA

J.G. Verwer, J.G. Blom, E.J. Spee
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

G.R. Carmichael
Center for Global and Regional Environmental Research
The University of Iowa, Iowa City, IA 52246, USA

Abstract

In the numerical simulation of atmospheric transport-chemistry processes, a major task is the integration of the stiff systems of ordinary differential equations describing the chemical transformations. It is therefore of interest to systematically search for stiff solvers which can be identified as close to optimal for atmospheric applications. In this paper we continue our investigation from [20] and compare eight solvers on a set of seven box-models used in present day models. The focus is on Rosenbrock solvers. These turn out to be very well suited for our application when they are provided with highly efficient sparse matrix techniques to economize on the linear algebra. Two of the Rosenbrock solvers tested are from the literature, viz. RODAS and ROS4, and two are new and developed for this benchmark, viz. RODAS3 and ROS3.

AMS Subject Classification (1991): Primary: 65L05. Secondary: 80A30, 65F05.
CR Subject Classification (1991): G1.7
Keywords & Phrases: Atmospheric chemistry, Air pollution modeling, Numerical stiff ODEs, Rosenbrock Methods, Sparsity.
Note: This work has also been released in the report series 'Reports on Computational Mathematics' of the University of Iowa (1996). For further publication the manuscript has been submitted to the journal Atmospheric Environment.

1. Introduction

To better understand the transport and fate of trace gases and pollutants in the atmosphere, comprehensive air quality models have been developed. For their numerical solution, very often the operator splitting approach is followed. A major computational task is then the numerical integration of the stiff ODE (ordinary differential equation) systems describing the chemical transformations. This integration must be carried out repeatedly at all spatial grid points for all split intervals chosen, so that model runs readily require an enormous amount of integrations. It is therefore of interest to systematically search for stiff ODE solvers which for atmospheric applications can be identified as close to optimal. In this paper we continue our search from [20], where a large number of box-model tests were carried out with nine solvers. Among these were dedicated explicit methods and general purpose solvers from the numerical stiff ODE field, all provided with sparse matrix techniques to economize on the numerical algebra operations. Three main conclusions were drawn in [20]:

- All sparse general solvers work quite efficiently on all test problems, although their ranking relative to each other depends on the test problem. Compared were the BDF solvers VODE [2] and LSODES [13], the Runge-Kutta solver SDIRK4 [11] and the Rosenbrock solver RODAS [11].
- TWOSTEP [27, 29] is by far the best within the class of dedicated explicit methods. It outperforms a number of QSSA solvers, often by a wide margin. However, it is in general less efficient than sparse implicit solvers. The code is advocated for gas-phase problems only and, like all other dedicated explicit solvers tested, not capable of treating gas-liquid phase chemistry.
- Sparse RODAS is competitive to all solvers tested and often is the fastest for low to moderate accuracies.

RODAS partly owes its competitiveness to its one-step nature. This is important in view of the large number of restarts carried out in the box-model runs. Restarts must be considered because the solvers are examined for application in an operator splitting approach. The multistep BDF (Gear) codes are then less attractive since their growth in step size is limited by stability considerations.

Our experience with RODAS is in line with results from [11], where for a number of stiff ODEs from other applications RODAS was shown to be competitive with other solvers for low to modest accuracies. Because for atmospheric applications the greatest interest lies in high efficiency for low accuracy (two figures at most), it is worthwhile to continue our search within the class of Rosenbrock methods. Thus, the aim of this paper is to assess whether other Rosenbrock solvers can be found which, for our specific purpose, constitute an improvement over RODAS in terms of efficiency.

The paper is organized as follows. In Section 2 we briefly review our test set from [20] and describe a new test problem. This test problem is also solved with the EBI method proposed in [17]. Section 3 contains a brief introduction to Rosenbrock methods, put together for the convenience of readers from the atmospheric research community. An appendix to this section is added for those readers who wish to learn more on Rosenbrock methods. In Section 4 we discuss all eight solvers which were tested. These include the two Rosenbrock solvers RODAS and ROS4 from [11] and two new Rosenbrock solvers which were developed for this benchmark, viz. RODAS3 and ROS3. The special purpose solver EBI from [17] was applied to the first test problem only, since it is dependent on the chemical mechanism. For the purpose of a wider comparison we also present results for the extrapolation code SEULEX from [11] and for TWOSTEP and VODE. The latter two were also tested in [20]. Section 5 describes the set up of the experiments and Section 6 contains all the test results. The final Section 7 summarizes the main conclusions.

To enable interested readers to further extend this benchmark comparison using their own solvers, as well as to extend our problem set with other challenging example problems from atmospheric chemistry, all the software we have used for the problems and the solvers have been put on the ftp-site [9].

2. The benchmark problems

The test set used in this paper consists of seven box-model problems. Except for number one, i.e. Problem A, all remaining problems, i.e. Problems B - G, are identical to those used in [20]. To save space we therefore present B - G only very briefly and refer to [20] for a complete description of these models. All problems were run for five days. This time interval is sufficiently large for taking into account several diurnal cycles of the photochemical reactions. The five day interval is split up in 120 one hour subintervals for B - G and in 180 forty minutes subintervals for A, while at the end of each subinterval the integration is interrupted and restarted, in accordance with the operator splitting approach. For all our test problems the unit of time is seconds and the unit for the concentrations is number of molecules per cm^3. All problems were uniformly coded in FORTRAN [1] using the symbolic

[1] Except for problem A, for which we have used an EBI implementation we obtained from [6].

preprocessor KPP [4]. This uniformity is important for a meaningful intercomparison.

Problem A*: The* TM*k model* The problem was borrowed from [6, 7]. It describes the reduced $CH_4/CO/HO_x/NO_x$ chemistry and is used in the global dispersion model TMk [12]. It consists of 36 reactions between 18 species of which 2 were held fixed, namely H_2O and O_2. Since new values of the photolysis rates are available every 40 minutes, we split accordingly the five day period. The simulated conditions correspond to a polluted air parcel in summer time, at 45 degrees north latitude and at ground level (pressure = 1000 mbar). We have included emissions of NO at a constant level of 10^6 mlc/cm^3/s. More information about this model can be found in [7]. We note that for this small problem (17 components) the exploitation of sparsity results in limited benefits. The Jacobian matrix has 90 nonzero entries and 93 after the factorization.

Problems B *and* C*: The* CBM-IV *model* These are based on the Carbon Bond Mechanism IV [10] consisting of 32 chemical species involved in 70 thermal and 11 photolytic reactions. Test problem B describes an urban scenario and simulates a heavily polluted atmosphere. Test problem C describes a rural atmosphere.

Problems D *and* E*: The* AL *model* Problems D and E employ the kinetic mechanism that is presently used in the STEM-II model [3], consisting of 84 non-constant chemical species involved in 142 thermal and 36 photolytic reactions. The mechanism, based on the work of [1] and [15], can be used to study the chemistry of both highly polluted (e.g., near urban centers) and remote (e.g., marine) environments. Problem D describes an urban scenario and problem E a rural one. The simulated conditions are identical to those employed in problems B and C, respectively.

Problem F*: A stratospheric model* This test problem is based on the chemical mechanism that was used in the NASA HSRP/AESA stratospheric models intercomparison. The initial concentrations and the values of the rate constants follow the NASA region A scenario. There are 34 non-constant species involved in 84 thermal and 25 photolytic reactions. No emissions were prescribed.

Problem G*: A wet model* The wet model contains 65 non-constant species involved in 77 thermal and 11 photolytic gas-phase chemical reactions, 39 liquid-phase chemical reactions and 39 gas-liquid mass transfer reactions. The gas-phase mechanism is based on CBM-IV, while the liquid-phase mechanism is based on a chemical scheme the authors obtained from [16]. All dedicated explicit solvers tested in [20] failed on this problem.

3. Rosenbrock methods

This section is devoted to a brief introduction to Rosenbrock methods, put together for the convenience of readers from the atmospheric research community. Part of the notation has been adopted from [11], where Rosenbrock methods are described in much greater detail (Sections IV.7, IV.10 and VI.3). An introductory appendix has been added for those readers who wish to learn more about the theory behind Rosenbrock methods.

3.1 The integration formula

Rosenbrock methods are usually considered in conjunction with stiff ODE systems in the autonomous form

$$\dot{y} = f(y), \quad t > t_0, \quad y(t_0) = y_0. \tag{3.1}$$

This places no restriction since every non-autonomous system $\dot{y} = f(t, y)$ can be put in the form (3.1) by treating time t also as a dependent variable, i.e. by augmenting the system with the equation $\dot{t} = 1$. In atmospheric applications it is often the case that the reaction coefficients are held constant on each split step interval; the chemical rate equations obtained this way are in autonomous form.

Usually stiff ODE solvers use some form of implicitness in the discretization formula for reasons of numerical stability. The simplest implicit scheme is the backward Euler method

$$y_{n+1} = y_n + hf(y_{n+1}), \tag{3.2}$$

where $h = t_{n+1} - t_n$ is the step size and y_n the approximation to $y(t)$ at time $t = t_n$. Since y_{n+1} is defined implicitly, this numerical solution itself must also be approximated. Usually some modification of the iterative Newton method is used, again for reasons of numerical stability. Suppose that just one iteration per time step is applied. If we then assume that y_n is used as the initial iterate, the following numerical result is found

$$\begin{aligned} y_{n+1} &= y_n + k, & \text{(3.3a)} \\ k &= hf(y_n) + hJk, & \text{(3.3b)} \end{aligned}$$

where J denotes the Jacobian matrix $f'(y_n)$ of the vector function f.

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 [18] 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 [11] for a so-called s-stage Rosenbrock method, is

$$\begin{aligned} y_{n+1} &= y_n + \sum_{i=1}^{s} b_i k_i, & \text{(3.4a)} \\ k_i &= hf(y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j) + hJ \sum_{j=1}^{i} \gamma_{ij} k_j, & \text{(3.4b)} \end{aligned}$$

where 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. An introduction on the properties of consistency, stability and stiff accuracy for Rosenbrock methods is presented in an appendix.

For a reason explained later, the coefficients γ_{ii} are taken equal for all stages, i.e. $\gamma_{ii} = \gamma$ for all $i = 1, \ldots, k$. For $s = 1$, $\gamma = 1$ the above linearized implicit Euler formula is recovered. For the non-autonomous system $\dot{y} = f(t, y)$, the definition of k_i is changed to

$$k_i = hf(t_n + \alpha_i, y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j) + \gamma_i h^2 \frac{\partial f}{\partial t}(t_n, y_n) + hJ \sum_{j=1}^{i} \gamma_{ij} k_j,$$

where

$$\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}, \quad \gamma_i = \sum_{j=1}^{i} \gamma_{ij}.$$

Like Runge-Kutta methods, Rosenbrock methods successively form intermediate results

$$Y_i = y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j, \quad 1 \le i \le s, \tag{3.5}$$

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 put $J = 0$, a classical explicit Runge-Kutta method results.

Rosenbrock methods are attractive for a number of reasons. Like fully implicit methods, they preserve 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. They can be developed to possess optimal linear stability properties for stiff problems. 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.

3.2 Reducing computational costs

Each time step requires an evaluation of the Jacobian J, s matrix-vector multiplications with J and, assuming that $\gamma_{ii} = \gamma$, s solutions of a linear system with (the same) matrix $I - \gamma h J$, accompanied with s derivative evaluations. The multiplications with J are easily avoided in the actual implementation by a simple transformation (see Section IV.7 of [11]). Still, stepwise counted, the computational costs for a Rosenbrock method are considered to be high compared to the costs of, say, a linear multistep method of the BDF type. In particular, the Jacobian update and the solution of the s linear systems, requiring one matrix factorization (LU-decomposition) and s backsolves (forward-backward substitutions) typically account for most of the CPU time used by a Rosenbrock method.

Sparsity For large atmospheric chemistry models the number of zeroes in J readily amounts to $\approx$ 90%. This high level of sparsity can be exploited to significantly reduce the costs of the linear algebra calculations. For this task we use the symbolic preprocessor KPP [4]. KPP prepares a sparse matrix factorization with only a minimal fill-in (see Table 1 in [20]) and delivers a FORTRAN routine for the backsolve without indirect addressing. Altogether this means that the numerical algebra is handled very efficiently. The sparse matrix technique implemented in KPP is based on a diagonal Markowitz criterion (see [4, 20, 21] for more details).

Approximate Jacobians It is conceivable to attempt to further reduce the numerical algebra costs through an approximate Jacobian.

- One possibility is to use a time-lagged Jacobian $J = f'(y_{n+\eta})$ where $\eta = 0, -1, \ldots$ such that $n+\eta$ is constant. If we define J this way, and in addition keep h fixed, then $I - \gamma h J$ is a constant matrix during the number of times that the parameter η is decreased; hence one can advance several time steps using the same LU-decomposition. The derivation of order conditions (which circumvents the order reduction associated with the time-lagging of the Jacobian) can be found in [25, 26]. Since the exact Jacobians are used, conservation properties will still be maintained.
- Replacing J by a matrix with a simpler structure, say a matrix of higher sparsity, may result in further savings in linear algebra costs, but will destroy the conservation properties. Also, the number of order conditions will significantly increase (see the W-methods of [23]).
- One can devise methods based on a partitioning of the species into slow and fast ones where part of the entries of J is put to zero. This approach does not maintain conservation properties either and adds the problem of devising a good partitioning strategy.

Although in this paper only exact Jacobians are considered, we plan to examine the above ideas and the possible benefits of approximate Jacobians in a future investigation.

3.3 Step size control

General purpose stiff ODE solvers normally adapt the step size in an automatic manner to enable small step sizes at times when the solution gradients are large and large step sizes when solution gradients are small. For Runge-Kutta solvers an effective and simple step size control can be based on a so-called embedded formula

$$\tilde{y}_{n+1} = y_n + \sum_{i=1}^{s} \tilde{b}_i k_i,$$

which uses the already computed increment vectors k_i. The approximation $\tilde{y}_{n+1}$ thus differs only in the choice of the weights $\tilde{b}_i$ and hence is available at no extra costs. Usually, the weights are chosen such that the order of consistency of $\tilde{y}_{n+1}$ is $\tilde{p} = p - 1$, if p is the order of y_{n+1}. This suggests to use the difference vector $Est = \tilde{y}_{n+1} - y_{n+1}$ as a local error estimator. In what follows we will denote the order of such a pair of formulas by $p(\tilde{p})$. All the Rosenbrock solvers (RODAS, RODAS3, ROS4 and ROS3) use embedded formulas to estimate the local error.

The specific step size strategy goes as follows. Let m denote the dimension of the ODE system. Let $Tol_k = atol + rtol\,|y_{n+1,k}|$, where $atol$ and $rtol$ represent a user-specified absolute and relative error tolerance and $y_{n+1,k}$ the k-th component of y_{n+1}. Tolerances may differ componentwise, but are here taken equal for all components for simplicity of testing. Denote

$$Err = \sqrt{\frac{1}{m}\sum_{k=1}^{m}\left(\frac{Est_k}{Tol_k}\right)^2}.$$

The integration step is accepted if $Err < 1$ and rejected otherwise and redone. The step size for the new step, both in the rejected and accepted case, is estimated by the usual step size prediction formula

$$h_{\text{new}} = h \,.\, \min\left(10,\, \max\left(0.1,\, 0.9/(Err)^{1/(\tilde{p}+1)}\right)\right).$$

At the first step after a rejection, the maximal growth factor of 10 is set to 1.0. Further, h is constrained by a minimum h_{min} and a maximum h_{max} and at any start of the integration for each operator-split interval we begin with a starting step size $h = h_{\text{start}}$. A rejection of the first step is followed by a ten times reduction of h. These step size constraints will be specified later. Because the maximal growth factor is equal to 10, the step size adjusts very rapidly and quickly attains large values if the solution is sufficiently smooth and $h = h_{\text{start}}$ is chosen small.

4. THE SOLVERS

In this section we list all solvers which have been tested. The solvers RODAS3 and ROS3 are new. For these we give the defining formula coefficients. All other solvers are existing ones and are described only briefly. The Rosenbrock solvers have order of consistency 3 or 4. Preliminary experiments with two second order solvers, based on Method III from [24] and on the complex-valued method from [18] (advocated in [19]) gave disappointing results.

RODAS This Rosenbrock solver from [11] is based on a stiffly accurate pair of order 4(3). Both formulas are L-stable. The number of stages s equals six and also six derivative evaluations and six backsolves are used. In [20] RODAS was one of the best solvers tested.

ROS4 This Rosenbrock solver is also taken from [11]. It implements a number of 4-stage 4(3) pairs which all require four derivative evaluations and four backsolves. Hence, per step ROS4 is somewhat cheaper than RODAS. However, in [11] a comparison is presented favouring RODAS, which is attributed to the stiff accuracy property (the methods of ROS4 are not stiffly accurate). We have tested its L-stable version (see Table 7.2, [11]) and found that generally its performance was very close to that of ROS3 and RODAS3. We therefore decided to omit presenting results for ROS4.

RODAS3 The third Rosenbrock solver was designed along the same principles as RODAS. It is based on a stiffly accurate, embedded pair of order 3(2). The number of stages is $s = 4$, requiring four backsolves but only three derivative evaluations are used. Hence per step it needs less work than RODAS, but it is one order lower. We have selected this pair since we aim at optimal efficiency for low accuracies. To the best of our knowledge, this pair of formulas has not yet been proposed elsewhere. The coefficients α_{ij} and γ_{ij} are

$$(\alpha_{ij}) = \begin{pmatrix} 0 & & & \\ 0 & 0 & & \\ 1 & 0 & 0 & \\ 3/4 & -1/4 & 1/2 & 0 \end{pmatrix}, \quad (\gamma_{ij}) = \begin{pmatrix} 1/2 & & & \\ 1 & 1/2 & & \\ -1/4 & -1/4 & 1/2 & \\ 1/12 & 1/12 & -2/3 & 1/2 \end{pmatrix},$$

and the weights are

$$(b_i) = \begin{pmatrix} 5/6 & -1/6 & -1/6 & 1/2 \end{pmatrix}, \quad (\tilde{b}_i) = \begin{pmatrix} 3/4 & -1/4 & 1/2 & 0 \end{pmatrix}.$$

Both formulas are L-stable. Observe that the embedded one is defined by the final intermediate approximation Y_4.

ROS3 The fourth Rosenbrock solver is based on an embedded pair of order 3(2) and is also new. The number of stages is $s = 3$ involving three backsolves and two derivative evaluations. The third order method is L-stable and the embedded second order method is strongly A-stable ($R(\infty) = 0.5$). The stiff accuracy property is not valid for ROS3. The method was constructed under the design criteria: order three, L-stability for both the stability function and the internal stability functions, and a strongly A-stable second order embedding. The internal stability functions are associated with the intermediate approximations (3.5). Imposing stability for these internal functions was advocated in [24] as a means to improve a Rosenbrock method for strongly nonlinear stiff problems. We note in passing that if the order of consistency equals 3 and $s = 3$, then the requirement of L-stability prevents the existence of an L-stable second order embedding. The coefficients are:

$$\begin{aligned}
\gamma &= 0.43586652150845899941601945119356 \\
\gamma_{21} &= -0.19294655696029095575009695436041 \\
\gamma_{32} &= 1.74927148125794685173529749738960 \\
b_1 &= -0.75457412385404315829818998646589 \\
b_2 &= 1.94100407061964420292840123379419 \\
b_3 &= -0.18642994676560104463021124732829 \\
\tilde{b}_1 &= -1.53358745784149585370766523913002 \\
\tilde{b}_2 &= 2.81745131148625772213931745457622 \\
\tilde{b}_3 &= -0.28386385364476186843165221544619
\end{aligned}$$

The remaining coefficients are $\alpha_{21} = \alpha_{31} = \gamma$ and $\alpha_{32} = \gamma_{31} = 0$.

VODE This solver from [2] is a general purpose BDF Gear code and can be regarded as a successor of LSODE [13], which is popular in the field of atmospheric chemistry as a reference code. In [20] VODE performed satisfactorily and we include it again for comparison with the Rosenbrock solvers. VODE uses the same sparsity routines as the Rosenbrock solvers.

TWOSTEP This solver from [27, 28, 29] is based on the second order BDF formula and uses, instead of the usual modified Newton method, Gauss-Seidel or Jacobi iteration for approximately solving the implicit BDF relations. In the tests of this paper only Gauss-Seidel iteration is used. It was developed as a special purpose, explicit solver for atmospheric chemistry problems. In [20] it outperforms a number of solvers based on the QSSA approach. We include it again for comparison with the Rosenbrock solvers. The same implementation as in [20] is used, which always performs two Gauss-Seidel iterations and automatically adjusts the step size.

SEULEX The solver SEULEX is also taken from [11]. It bears a relationship with the Rosenbrock solvers, as it builds up a solution from the (non-autonomous) linearly implicit Euler method, i.e., $y_{n+1} = y_n + (I - hJ)^{-1} hf(t_n, y_n)$, by Richardson extrapolation. The use of this Euler method in an extrapolation code for stiff ODEs was first suggested in Deuflhard [8]. A rule of thumb is that the virtue of extrapolation manifests itself most clearly when high accuracy is required (see also [11]). We have included SEULEX in our benchmarking as the extrapolation approach is mentioned by Zlatev [30] (see Section 3.4.3) as a viable one for atmospheric ODE problems, although no results seem to have been reported yet. The same sparse linear algebra as used for the other solvers was implemented. The extrapolation sequence defined by iwork(4) = 4 was used. This sequence was found to work well for our application. Other settings are given default values.

EBI This method (described in detail in [17]) is based on the Euler Backward Implicit formula (3.2). Its main feature is that, instead of using Newton's method, the implicit solution is approximated through a semi-analytical, problem dependent iteration process. This process groups species together which allow an exact solution of the implicit equations after putting part of them at the old time level. Species equations which do not fit in an appropriate grouping are treated with a form of Jacobi iteration. Satisfactory results are reported [17] for different scenario's based on the CBM-IV mechanism. The approach can also be applied when using higher BDF methods since use of these implicit methods leads to a similar system of equations, but a considerable drawback is that the iterative solution method is adapted to the particular chemistry scheme. We therefore have tested the method only for the TMk model, using an implementation obtained from [6]. This implementation contains no local error control mechanism so that constant step sizes are taken.

5. Set up of experiments

Accuracy All tests were carried out in the same way as in [20]. The numerical results were compared to a very accurate reference solution (given by RADAU5, $rtol = 10^{-12}$, componentwise set $atol$) using a temporal modified root mean square norm of the relative error. With the reference solution y and the numerical solution $\hat{y}$ available at $\{t_n = t_0 + n\Delta t,\ 0 \le n \le N\}$, where n is associated with the end points of the N operator-split subintervals, we first compute for each species k

$$ER_k = \sqrt{\frac{1}{|\mathcal{J}_k|} \cdot \sum_{n \in \mathcal{J}_k} \left| \frac{y_k(t_n) - \hat{y_k}(t_n)}{y_k(t_n)} \right|^2},$$

where $\mathcal{J}_k = \{0 \le n \le N : \ y_k(t_n) \ge a\}$. This value is then represented in the plots through the number of significant digits for the maximum of ER_k, defined by

$$SDA = -\log_{10}(\max_k ER_k). \tag{5.6}$$

Note that if the set $\mathcal{J}_k$ is empty for a chosen threshold a, the value of ER_k is neglected. This threshold factor serves to eliminate chemically meaningless large relative errors for concentration values smaller than a mlc/cm^3 in the error measure. We used $a = 10^6$ mlc/cm^3 for all tropospheric problems and $a = 10^4$ mlc/cm^3 for the stratospheric one. Additional experiments performed with $a = 1$ mlc/cm^3 led to nearly the same conclusions. Observe that $SDA = 2$ means 1% accuracy in the error measure used. In discussing the results presented in the next section we focus on this accuracy level.

Timing The answer to the question of which method is "the fastest" may depend also on the machine. In order to measure the influence of the hardware on the relative performance of integrators we have performed all the numerical experiments on two completely different architectures, namely a HP-UX 935 A workstation (double precision, ≈ 14 digits) and a Cray C98 (scalar mode, single precision, ≈ 14 digits); in addition, some of the experiments were also repeated on a SGI workstation (double precision, ≈ 14 digits). Somewhat to our surprise, very similar results were found; as a consequence, in what follows only the HP work-precision diagrams are presented. We plot the SDA values against efficiency, i.e., the measured CPU times on a logarithmic scale in unit seconds.

Steering parameters For all solvers important steering parameters are h_{start}, h_{min}, h_{max} and the local error tolerances *atol*, *rtol*. A user-specified choice for h_{min} is important. Without a prescribed minimum, step sizes can result as small as the shortest time constants, sometimes even ≈ 10^{-8} to 10^{-9} sec. Step size values close to these extremely short time constants are redundant, since the minimal time constants of importance for photochemical models lie between 1 sec and 1 min, approximately. On the temporal scale of interest, species with a smaller time constant quickly reach their (solution dependent) steady state when they are perturbed. We have prescribed the following values for h_{min} and h_{start} which are imposed for all solvers (except EBI): for Problems A-E, $h_{\text{min}} = 0.1$ sec and $h_{\text{start}} = 60$ sec; for Problem F, $h_{\text{min}} = h_{\text{start}} = 0.001$ sec; and for Problem G, $h_{\text{min}} = h_{\text{start}} = 0.0001$ sec. The maximal value h_{max} is less important and was not defined in our tests. Finally, for all problems and all solvers we have prescribed the absolute tolerance value $atol = 0.01$ mlc/cm^3 along with a number of relative tolerance values *rtol* such that effectively relative local error control is imposed. The different data points in the plots for a given method correspond to these relative tolerances.

Reaction coefficients In practice the rate coefficients are implemented in two ways, either as time-continuous functions or as functions piecewise constant per operator-split subinterval. The time-continuous function implementation of the thermal rate coefficients may lead to a large number of exponential function evaluations per time step, which are very costly (with Rosenbrock methods these calculations can be as expensive as the sparse matrix factorization). Since for the actual practice true time dependency seems redundant, we have used piecewise constant rate coefficients per operator-split subinterval (temperature and solar angle frozen using values halfway). Observe that in [20] time-continuous values were used. For the CBM-IV model an accuracy-efficiency plot will also be presented for time-continuous values. This provides us with the possibility to examine whether the solvers behave differently for the time-continuous and piecewise constant case.

6. Results and illustrations

6.1 Problem A: The TMk model

The work precision diagram is given in Figure 1. Results are presented for all the solvers discussed above, including EBI. The EBI results are obtained with a sequence of fixed step sizes of which the largest is ≈ 13.3 min. and the smallest 0.5 min. The number of iterations within EBI was in all runs equal to 8 (cf. [6]). The results show that the variable step size Rosenbrock solvers are clearly superior to all others for 1% accuracy. Noteworthy is that EBI and TWOSTEP are fast for very low accuracies (around 10% say). SEULEX appears to be faster than VODE, but slower than the Rosenbrock codes. However, the gap between these solvers decreases for higher accuracies; in fact SEULEX will take the lead for 5 or more accurate digits. Among the Rosenbrock codes, RODAS3 and ROS3 have similar performance in the low accuracy domain; they are followed closely by RODAS.

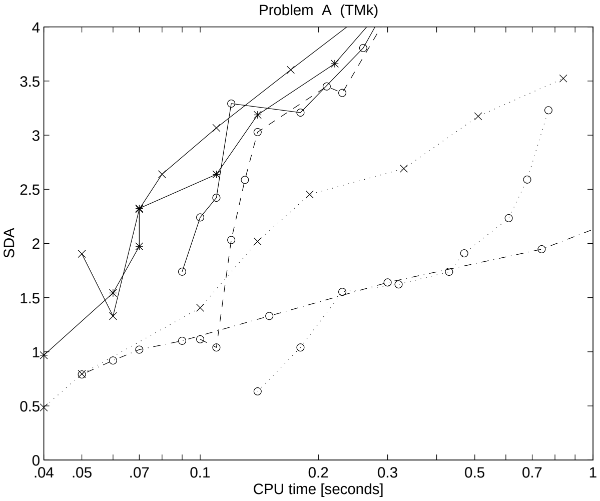

Figure 1: Work-precision diagram for test problem A (TMk): Sparse RODAS3 (solid with "$*$"), Sparse ROS3 (solid with "x"), Sparse RODAS (solid with "o"), TWOSTEP SEIDEL (dots with "x"), Sparse VODE (dots with "o"), Sparse SEULEX (dashed with "o") and EBI (dash-dots with "o").

6.2 Problems B *and* C*: The* CBM-IV *model*

In Figure 2 the results for test problems B and C are presented. For the rural problem all Rosenbrock solvers perform equally well, followed by SEULEX, while VODE and TWOSTEP fall behind. This also holds for the urban problem, but now a distinction exists between the Rosenbrock solvers and SEULEX. Up to about 3 digits RODAS3 and ROS3 perform best. For accuracies higher than 3 digits RODAS takes over.

From the numerical point of view it is of interest to also solve a problem where the reaction coefficients are time continuous (non-autonomous problem). Figure 3 shows results for Problems B and C. These should be compared with the results for the associated problem with coefficients piecewise constant per operator-split interval. In the urban case RODAS3 and ROS3 are again the best up to 3 digits accuracy followed by SEULEX and RODAS. TWOSTEP and VODE perform more or less as in the piecewise constant case, delivering 1% accuracy in about twice the CPU time needed by the Rosenbrock codes. In the rural scenario the Rosenbrock codes and SEULEX perform similarly, all of them being again notably faster than the BDF candidates. The results for the non-autonomous problem are similar to the results for the autonomous variant.

6.3 Problems D *and* E*: The* AL *model*

For problems D and E the results are given in Figure 4. It is interesting to compare code performances to those obtained for the CBM-IV model since the same urban and rural scenario's are simulated. They differ, however, in the number of species and reactions, the AL model being considerably larger. For the urban problem RODAS3 and ROS3 are again the fastest, up to 3 digits, while for higher accuracies RODAS becomes better. SEULEX now performs somewhat less than for the CBM-IV model, whereas TWOSTEP is notably better positioned. In the rural case RODAS is the best, but RODAS3, ROS3,

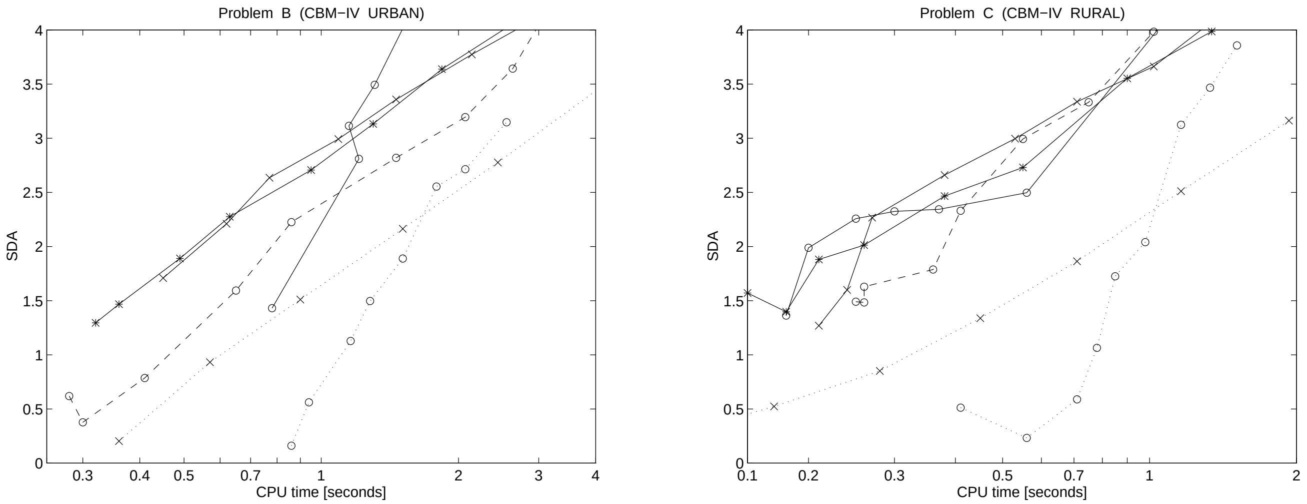

Figure 2: Work-precision diagram for test problems B and C (CBM-IV): Sparse RODAS3 (solid with "$*$"), Sparse ROS3 (solid with "x"), Sparse RODAS (solid with "o"), TWOSTEP SEIDEL (dots with "x"), Sparse VODE (dots with "o") and Sparse SEULEX (dashed with "o").

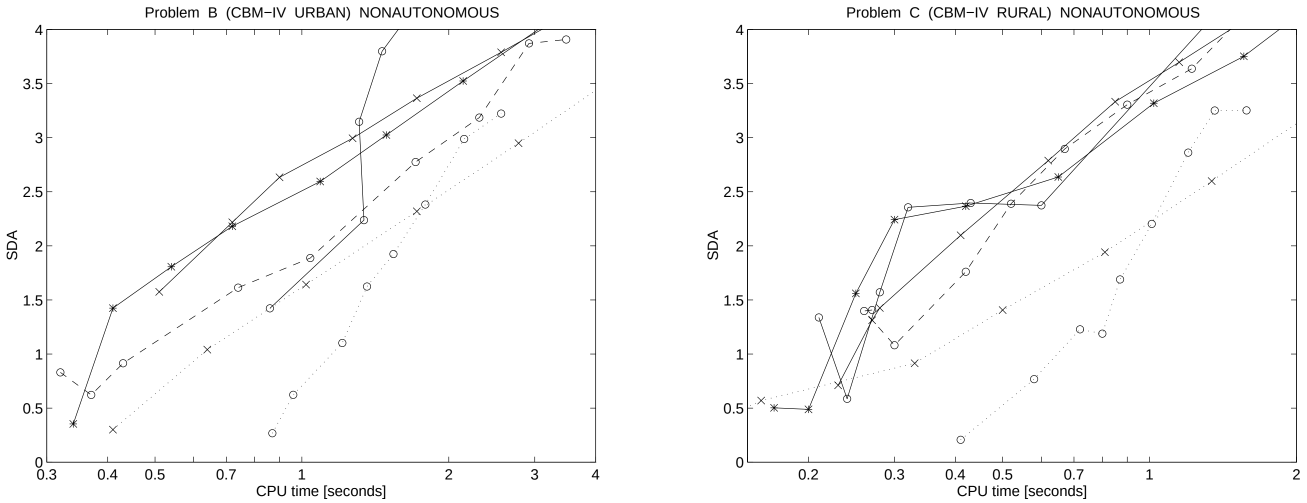

Figure 3: Work-precision diagram for test problems B and C (CBM-IV), the non-autonomous version: Sparse RODAS3 (solid with "$*$"), Sparse ROS3 (solid with "x"), Sparse RODAS (solid with "o"), TWOSTEP SEIDEL (dots with "x"), Sparse VODE (dots with "o") and Sparse SEULEX (dashed with "o").

SEULEX and TWOSTEP follow it very closely. Both in the rural and urban case VODE falls behind. Notable is the close performance of ROS3 and RODAS3. As a general conclusion, Rosenbrock codes are again superior to the BDF ones. The better relative positioning of TWOSTEP (as compared to the CBM-IV cases) is most likely due to the increased number of species in AL. This fact suggests that, for very large kinetic systems (say, with more than 100 components) explicit solvers like TWOSTEP may become competitive.

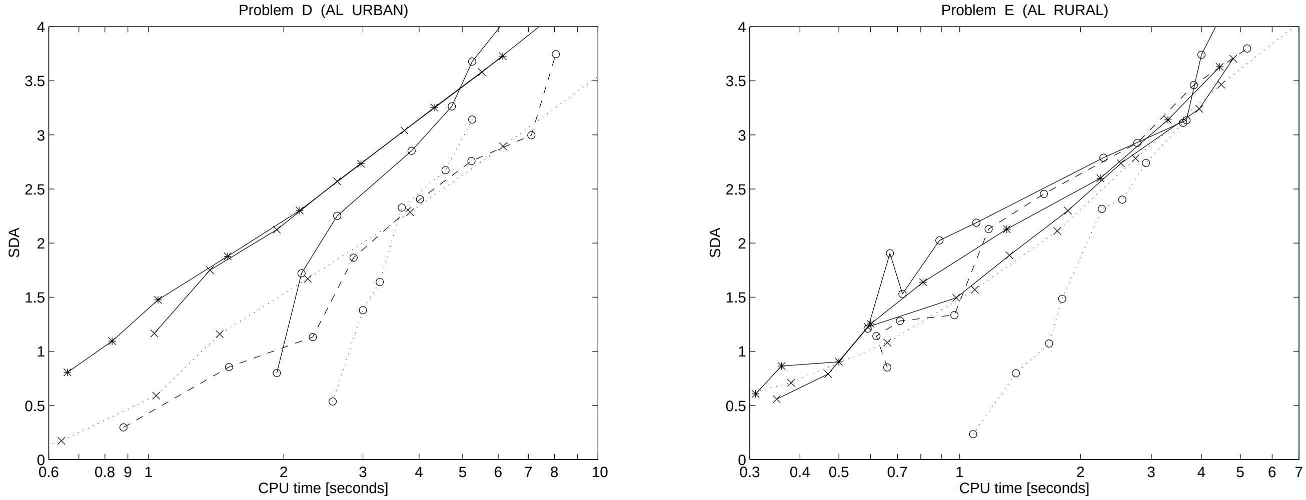

Figure 4: Work-precision diagram for test problems D and E (AL): Sparse RODAS3 (solid with "*"), Sparse ROS3 (solid with "x"), Sparse RODAS (solid with "o"), TWOSTEP SEIDEL (dots with "x"), Sparse VODE (dots with "o") and Sparse SEULEX (dashed with "o").

6.4 Problem F*: The stratospheric model*

The work-precision diagram given in Figure 5 again reveals a very good performance of the Rosenbrock solvers compared to the other three. The higher order of accuracy of RODAS is again borne out and again notable is the close performance of ROS3 and RODAS3. VODE and SEULEX have similar performance, but are more than 2 times slower than the Rosenbrock codes in the 1% accuracy range. TWOSTEP follows at a large distance.

6.5 Problem G*: The wet model*

As pointed out in [20], this test problem is the most difficult one from the numerical point of view. The Jacobian $f'(y)$ of the derivative function (3.1) contains stiff eigenvalues for which the relation $\lambda_i \approx -L_i$ (with L_i the destruction term associated with species i) does not hold. Such eigenvalues are due to the rapid gas-liquid phase interactions and cannot be associated with certain species; for this reason, all the explicit solvers tested in [20] failed to efficiently integrate the WET model. As a consequence, in the present work TWOSTEP was not applied to this problem. The results plotted in Figure 6 for the other solvers are very much in line with those for the stratospheric problem. In the low accuracy range the Rosenbrock family has the lead again, the performances of RODAS, RODAS3 and ROS3 being very close to each other. SEULEX is about three times slower for 2 accurate digits, but seems to become the best for more than 4 digits; for higher accuracies, VODE changes slope and is not competitive.

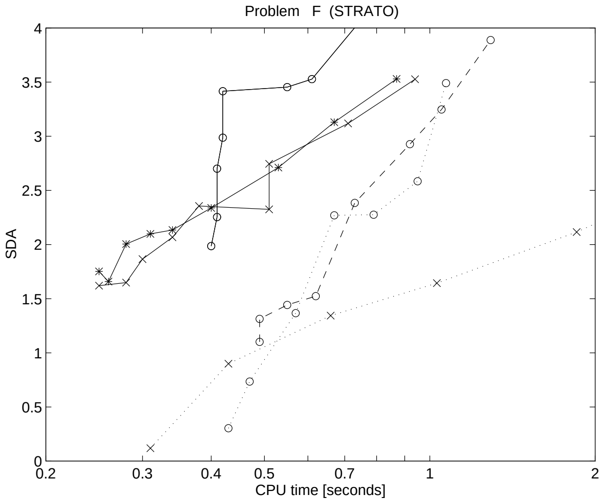

Figure 5: Work-precision diagram for test problem F (STRATO): Sparse RODAS3 (solid with "∗"), Sparse ROS3 (solid with "x"), Sparse RODAS (solid with "o"), TWOSTEP SEIDEL (dots with "x"), Sparse VODE (dots with "o") and Sparse SEULEX (dashed with "o").

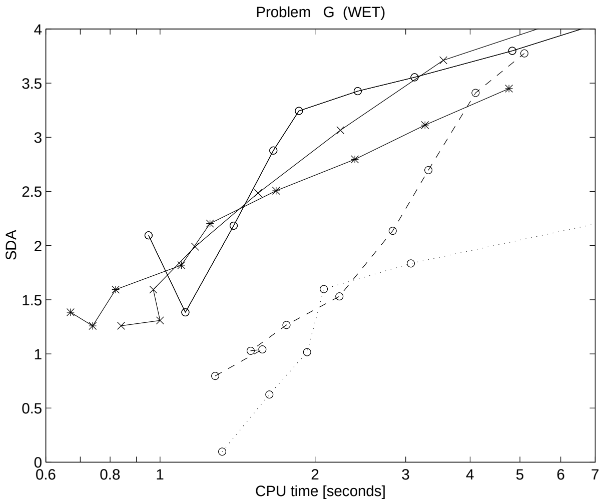

Figure 6: Work-precision diagram for test problem G (WET): Sparse RODAS3 (solid with "∗"), Sparse ROS3 (solid with "x"), Sparse RODAS (solid with "o"), Sparse VODE (dots with "o") and Sparse SEULEX (dashed with "o").

7. Overall conclusions and remarks

- For the accuracy range considered (up to 4 digits) it is almost always a sparse Rosenbrock solver which is the fastest. The relative ranking between the four sparse Rosenbrock solvers differs per problem, but only to a limited amount. As expected, for higher accuracies RODAS is generally the best. For lower accuracies of practical interest RODAS3, ROS3 and ROS4 are mostly competitive. In passing we note that our test results do not consistently show that the property of stiff accuracy is truly advantageous for nonlinear problems.

- The above conclusion about the computational speed of Rosenbrock methods is also supported by the comparison with the EBI method for Problem A and with the QSSA method (not presented here, but see [20]). In particular, RODAS3 is about 5 to 10 times faster than QSSA for 10% relative errors and 20 to 100 times faster for 1% relative errors.

- While box model tests are needed to select and develop promising ODE solvers, in real 3D transport-chemistry models other factors should be taken into account as well. Quite important is the length of the time step in the operator splitting, since this determines the number of restarts. Restarts are expensive and one-step methods have an advantage here over multistep methods. Also robustness and ease of use are important in 3D models, since a subtle tuning of the ODE code is cumbersome due to the large variety of conditions that will occur at different grid points. In this respect the Rosenbrock solvers are also very attractive. All tests confirm that they are easy to use and robust. We have to point out, though, that for several test problems very large values of *rtol* (≥ 0.1) caused some of the Rosenbrock solvers to drift away from the real solution.

- The answer to the question of which stiff integrator is "the best" for being used in air quality models depends on a multitude of factors, some of the most important being the specific chemical mechanism employed, the desired accuracy level and the hardware on which the code runs. In the present work we consider a variety of chemical models, we cover the whole range of accuracy levels of practical interest and run everything on two machines with completely different architectures. Since the Rosenbrock methods systematically perform best, our summarizing general impression is that for atmospheric chemistry problems the sparse Rosenbrock solvers are close to optimal. In particular, for low accuracies up to two or three figures RODAS3 is never disappointing. Even if RODAS3 would not be optimal in a certain setting, it will perform very close to "the optimal method" in that setting, say within 25% of CPU performance.

Although we have used utmost precaution in implementing the models and in testing the codes, still undiscovered errors and/or less optimal settings of user parameters may have affected part of the numerical results. The interested reader therefore is invited to repeat the experiments using our codes from [9] and to join us[2] in this benchmark activity.

Acknowledgements The authors gratefully acknowledge the following organizations for their financial support: A. Sandu and G.R. Carmichael were supported by the DOE under grant no DE-FG02-94ER 61855, J.G. Blom by the Dutch HPCN Program (TASC project HPCN for Environmental Applications), E.J. Spee by the RIVM (Dutch National Institute of Public Health and Environmental Protection, Project CIRK) and J.G. Verwer by Cray Research Inc. via the Stichting Nationale Computer Faciliteiten (National Computing Facilities Foundation, grant CRG 96.03, Project CIRK).

[2] Contact A. Sandu (sandu@cgrer.uiowa.edu) or J.G. Verwer (janv@cwi.nl).

References

1. R.D. Atkinson, D.L. Baulch, R.A. Cox, R.F.JR. Hampson, J.A. Kerr, and J. Troe. Evaluated kinetic and photochemical data for atmospheric chemistry. *Journal of Chemical Kinetics*, 21:115–190, 1989.
2. P.N. Brown, G.D. Byrne, and A.C. Hindmarsh. VODE: A variable coefficient ODE Solver. *SIAM J. Sci. Stat. Comput.*, 10:1038–1051, 1989.
3. G.R. Carmichael, L.K. Peters, and T. Kitada. A second generation model for regional-scale transport/chemistry/deposition. *Atmospheric Environment*, 20:173–188, 1986.
4. V. Damian-Iordache and A. Sandu. KPP - A symbolic preprocessor for chemistry kinetics - User's guide. Technical report, University of Iowa, Department of Mathematics, 1995.
5. K. Dekker and J.G. Verwer. *Stability of Runge-Kutta methods for stiff nonlinear differential equations.* North-Holland, 1984.
6. F. Dentener. Private communication, Institute for Marine and Atmospheric Research, University of Utrecht, The Netherlands, 1996.
7. F. Dentener. Heterogeneous chemistry in the troposphere. PhD Thesis, Institute for Marine and Atmospheric Research, University of Utrecht, The Netherlands, 1993.
8. P. Deuflhard. Recent progress in extrapolation methods for ordinary differential equations. *SIAM Review*, 27:505–535, 1985.
9. ftp.cgrer.uiowa.edu. Ftp site at the Center for Global and Regional Environmental Research, University of Iowa (cd pub/Ode_benchmark, mget *), 1996.
10. M.W. Gery, G.Z. Whitten, J.P. Killus, and M.C. Dodge. A photochemical kinetics mechanism for urban and regional scale computer modeling. *Journal of Geophysical Research*, 94:12925–12956, 1989.
11. E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems.* Springer-Verlag, Berlin, 1991.
12. M. Heimann. The global atmospheric tracer model TMk. Report No. 10, Deutches Klimarechnenzentrum (DKRZ), 1995.
13. A.C. Hindmarsh. *ODEPACK: A systematized collection of ODE solvers.* In: R.S. Stepleman (ed.), IMACS Trans. on Scientific Computation, Vol. 1, Scientific Computing, North Holland, Amsterdam, 1983.
14. M. van Loon. *Numerical methods in smog prediction.* PhD Thesis University of Amsterdam (will also be published as CWI Tract), 1996.
15. F.W. Lurmann, A.C. Loyd, and R. Atkinson. A chemical mechanism for use in long-range transport/acid deposition computer modeling. *Journal of Geophysical Research*, 91:10905–10936, 1986.
16. J. Matthijssen. Private communication, Laboratoire d'Aerologie OMP, Toulouse, France, 1995.
17. O. Hertel, R. Berkowicz, J. Christensen, and Ø. Hov. Test of two numerical schemes for use in atmospheric transport-chemistry models. *Atmospheric Environment*, 27A:2591–2611, 1993.
18. H.H. Rosenbrock. Some general implicit processes for the numerical solution of differential equations. *Comput. J.,* 5:329–330, 1963.
19. E.Yu. Dnestrovskaya, N.N. Kalitkin, and L.V. Kusmina. Lq-decreasing monotonic schemes with complex coefficients and applications to complicated PDEs. *Applied Numerical Mathematics,* 15:327–340, 1994.
20. A. Sandu, J.G. Verwer, M. van Loon, G.R. Carmichael, F.A. Potra, D. Dabdub and J.H. Seinfeld. Benchmarking stiff ODE solvers for atmospheric chemistry problems I: implicit versus explicit. CWI Report NM-R9603 and Report in Computational Mathematics no 85, Department of Mathematics,

The University of Iowa. In revised form accepted by *Atmospheric Environment*, 1996.

21. A. Sandu, F.A. Potra, V. Damian, and G.R. Carmichael. Efficient implementation of fully implicit methods for atmospheric chemistry. Technical report 79, University of Iowa, Department of Mathematics, 1995. In revised form accepted by *Journal of Computational Physics*, 1996
22. D. Simpson, Y. Andersson-Sköld, and M.E. Jenkin. Updating the chemical scheme for the EMEP MSC-W oxidant model: current status. Technical Report 2/93, EMEP MSC-W, The Norwegian Meteorological Institute, Oslo, 1993.
23. T. Steihaug, and A. Wolfbrandt. An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff differential equations. *Math. Comp.*, 33:521–534, 1079.
24. J.G. Verwer. S-stability properties for generalized Runge-Kutta methods. *Numer. Math.*, 27:359–370, 1977.
25. J.G. Verwer and S. Scholz. Rosenbrock methods and time-lagged Jacobian matrices. *Beiträge zur Numer. Math.*, 11:173–183, 1983.
26. J.G. Verwer, S. Scholz, J.G. Blom and M. Louter-Nool. A class of Runge-Kutta-Rosenbrock methods for solving stiff differential equations. *Z. Angew. Math. Mech.*, 63:13–20, 1983.
27. J.G. Verwer. Gauss-Seidel iterations for stiff ODEs from chemical kinetics. *SIAM Journal of Scientific Computing*, 15:1243–1250, 1994.
28. J.G. Verwer, and D. Simpson. Explicit methods for stiff ODEs from atmospheric chemistry. *Applied Numerical Mathematics*, 18: 413–430, 1995.
29. J.G. Verwer, J.G. Blom, M. van Loon, and E.J. Spee. A comparison of stiff ODE solvers for atmospheric chemistry problems. *Atmospheric Environment*, 30:49-58, 1996.
30. Z. Zlatev. *Computer Treatment of Large Air Pollution Models.* Kluwer Academic Publishers, 1995.

A. Appendix: Consistency and Stability of Rosenbrock methods

The performance of an integration method largely depends on its order of consistency and its stability properties. Again for the convenience of readers from the atmospheric research community, in this Appendix we will briefly discuss the consistency property for the Rosenbrock method, as well as some useful results from the linear stability theory. Also some attention will be paid to the notion of stiff-accuracy.

Consistency conditions The consistency conditions are found from a formal Taylor expansion of the local error. Let $y_{n+1} = E(y_n)$ be a compact notation for the Rosenbrock method. The difference

$$\delta_h(t) = E(y(t)) - y(t+h), \tag{A.7}$$

where y is the exact (local) solution of the ODE system $\dot{y} = f(y)$ passing through $y(t)$, is called the local error and the largest integer p for which

$$\delta_h(t) = O(h^{p+1}), \quad h \to 0,$$

is called the order of consistency. Hence $\delta_h(t)$ is the error after a single step from an exact solution, while the order reveals how rapidly $\delta_h(t)$ approaches zero for a decreasing step size. Assuming sufficient differentiability of y and f, the order p is determined by Taylor expanding the local error and equating to zero the resulting terms up to the p-th one. This leads to the so-called consistency conditions which are expressions in the formula coefficients. Satisfying these conditions gives the desired order p. While the expansion is technically complicated and the resulting conditions can become quite lengthy for a large p, the derivations are conceptually simple. For a maximum of four stages, the conditions for order $p \leq 3$ are:

$$p=1 \quad : \quad b_1 + b_2 + b_3 + b_4 = 1, \tag{A.8a}$$

$$p=2 \quad : \quad b_2\beta_2' + b_3\beta_3' + b_4\beta_4' = \frac{1}{2} - \gamma, \tag{A.8b}$$

$$p=3 \quad : \quad b_2\alpha_2^2 + b_3\alpha_3^2 + b_4\alpha_4^2 = \frac{1}{3}, \tag{A.8c}$$

$$: \quad b_3\beta_{32}\beta_2' + b_4(\beta_{42}\beta_2' + \beta_{43}\beta_3') = \frac{1}{6} - \gamma + \gamma^2, \tag{A.8d}$$

where

$$\beta_{ij} = \alpha_{ij} + \gamma_{ij}, \quad \alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}, \quad \beta_i' = \sum_{j=1}^{i-1} \beta_{ij}.$$

The conditions for $p \leq 5$ and general s can be found in Section IV.7 of [11].

Linear stability Let $\epsilon_n = y_n - y(t_n)$ denote the global error: the difference between the sought exact solution of the ODE system $\dot{y} = f(y)$ and the computed approximation. The global error at the forward time level $t = t_{n+1}$ can be seen to satisfy

$$\epsilon_{n+1} = E(\epsilon_n + y(t_n)) - E(y(t_n)) + \delta_h(t_n), \tag{A.9}$$

showing that this error consists of two parts: the local error (A.7), which is a functional of the exact solution, and the difference

$$E(\epsilon_n + y(t_n)) - E(y(t_n)),$$

where $E(\epsilon_n + y(t_n))$ represents the actual Rosenbrock step taken from the approximation $y_n = \epsilon_n + y(t_n)$ and $E(y(t_n))$ represents the hypothetical Rosenbrock step taken from the exact solution $y(t_n)$. This difference term reveals a dependence of ϵ_{n+1} on ϵ_n. For a proper functioning of the Rosenbrock method it is desirable that, in an appropriate norm $\| \ \|$,

$$\| E(\epsilon_n + y(t_n)) - E(y(t_n))\| \leq \| \epsilon_n\|, \tag{A.10}$$

because then the integration is stable in the sense that

$$\| \epsilon_{n+1}\| \leq \| \epsilon_n\| + \| \delta_h(t_n)\|.$$

This error inequality is elementary, but also fundamental for one-step integration methods. It simply shows that all local errors add up to the global error,

$$\| \epsilon_n\| \leq \sum_{j=0}^{n-1} \| \delta_h(t_j)\|,$$

if we assume that at the initial time t_0 the error $\epsilon_0 = 0$. From inserting $\delta_h(t_j) = O(h^{p+1})$, while assuming $h \to 0$ and $n \to \infty$ such that $t_n = nh$ is fixed, it follows that $\epsilon_n = O(h^p)$. By adding up all local errors one power of h is lost, resulting in a convergence order p.

If (A.10) does not hold, the global error can accumulate unboundedly. The integration is then unstable and of no practical use. Whereas for general nonlinear stiff ODEs from chemistry no stability analysis exists for Rosenbrock methods, their stability is well understood for stable, linear systems

$$\dot{y} = Jy, \tag{A.11}$$

with eigenvalues λ satisfying $\mathrm{Re}(\lambda) \leq 0$. From practical experience we know that linear stability often provides a satisfactory prediction of stability for nonlinear problems if J is interpreted as the Jacobian matrix $f'(y)$. This interpretation is based on a linearization argument [5, 11]. Applied to (A.11), the Rosenbrock method $y_{n+1} = E(y_n)$ reduces to the linear recursion

$$y_{n+1} = R(hJ)y_n, \tag{A.12}$$

where $R(hJ)$ is a matrix-valued rational function that approximates the matrix-valued exponential function e^{hJ}, being the solution operator of (A.11). By inserting (A.12) into the error equation (A.9), we obtain

$$\epsilon_{n+1} = R(hJ)\epsilon_n + \delta_h(t_n),$$

or, equivalently,

$$\epsilon_n = R^n(hJ)\epsilon_0 + \sum_{j=0}^{n-1} R^{n-1-j}(hJ)\delta_h(t_j),$$

where, as before, $n = 1, 2, \ldots$. We see that the demand of stability can now be expressed as boundedness of powers of $R(hJ)$, i.e.,

$$\| R^n(hJ)\| \leq C, \tag{A.13}$$

where C is a constant which is independent of n and hJ. This independence guarantees unconditional stability in the sense that no restrictions exist on the step size. Condition (A.13) holds if we require that the scalar rational function $R(z)$, which is called the stability function, satisfies $|R(z)| \leq 1$ for arbitrary $z = h\lambda$, $\mathrm{Re}(z) \leq 0$. This is the famous property of A-stability originally proposed by Dahlquist (see [11]). We note in passing that for our application we do not really need A-stability, since for atmospheric chemistry the eigenvalues of the Jacobian are always located in the neighbourhood of the real axis. So we actually need the boundedness property only near the negative half line.

We will impose the condition of L-stability, which in addition to A-stability, requires $R(\infty) = 0$. L-stability is known to lead to a somewhat more robust approach and better mimics the damping property of e^z for $\mathrm{Re}(z) \leq 0$. The property of L-stability is easily verified. The stability function R is found by applying the method to the scalar problem $\dot{y} = \lambda y$. This yields a rational function of the form

s	L-stability, $p \geq s-1$	L-stability, $p = s$
1		$\gamma = 1$
2	$(2-\sqrt{2})/2 \leq \gamma \leq (2+\sqrt{2})/2$	$\gamma = (2 \pm \sqrt{2})/2$
3	$0.18042531 \leq \gamma \leq 2.18560010$	$\gamma = 0.43586652$
4	$0.22364780 \leq \gamma \leq 0.57281606$	$\gamma = 0.57281606$

Table 1: Values of γ for L-stability.

$$R(z) = \frac{P(z)}{(1-\gamma z)^{s'}}, \tag{A.14}$$

where P is a polynomial of degree $s', s' \leq s$, and the degree of P is less than or equal to $s'-1$ if the stability function is to be L-stable. Mostly, s' is equal to the number of stages s, but s' can be smaller. In this paper we only consider methods for which $s' = s$.

Stability properties of rational functions of the type (A.14) have been studied extensively. For our purpose the following results are very useful. Suppose that the order of consistency p of the Rosenbrock method is also the order of consistency of R, i.e., p is the largest integer for which $R(z) = e^z + O(z^{p+1}), z \to 0$. For L-stable functions we then usually have $p = s$ or $p = s-1$. In both cases R is uniquely determined by γ. For the case $p = s-1$, L-stability holds for certain intervals for γ and if $p = s$ for one particular value of γ (see Section IV.6 and Table 6.4 in [11]). By way of illustration we list the values of γ for $1 \leq s \leq 4$ in Table 1.

Stiff accuracy Stiff accuracy is a property related to the Prothero-Robinson model problem

$$\dot{y} = \lambda(y - \phi(t)) + \dot{\phi}(t),$$

where ϕ is some known function. Its solution reads

$$y(t+h) = e^{\lambda h}(y(t) - \phi(t)) + \phi(t+h)$$

and if $\mathrm{Re}(\lambda h) \to -\infty$, the solution $y(t+h) \to \phi(t+h)$, irrespective the size of h. Prothero and Robinson have investigated under which conditions on the formula coefficients, implicit Runge-Kutta solutions mimic this property. Because, then a method can handle this particular transition to infinite stiffness in an accurate manner, which has been the main motivation for this test model (see [5, 11]). They proposed the term stiff accuracy for this phenomenon.

For the current test model, the global error recursion (A.9) reads

$$\epsilon_{n+1} = R(z)\epsilon_n + \delta_h(t_n),$$

where $\delta_h(t_n)$ depends in a certain way on $z = h\lambda$, h and ϕ. Hairer and Wanner [11] show, in Section IV.15, that for any consistent Rosenbrock method,

$$\delta_h(t_n) = O(h^2/z), \quad \text{for} \quad h \to 0,\ z \to \infty,$$

if

$$\alpha_{si} + \gamma_{si} = b_i \quad (i = 1, \ldots, s) \quad \text{and} \quad \alpha_s = 1. \tag{A.15}$$

Hence, the desired transition property holds for the local error and because (A.15) also implies $R(\infty) = 0$, this property holds for the global error as well. They therefore call a Rosenbrock method stiffly accurate if (A.15) holds.

For general nonlinear stiff problems the virtue of stiff accuracy is not so clear. In [11] it is argued that stiff accuracy is advantageous when solving stiff differential-algebraic systems with a Rosenbrock method (cf. Proposition 3.12, Sect. VI.3). For ODEs a similar argument exists which goes as follows. Suppose (A.15) holds. A straightforward computation then reveals the following relation between y_{n+1} and the final stage quantities k_s and Y_s,

$$k_s = hJy_{n+1} + hf(Y_s) - hJY_s. \tag{A.16}$$

Assuming that J is invertible, we may write

$$y_{n+1} = Y_s - (hJ)^{-1}(hf(Y_s) - k_s), \tag{A.17}$$

which is the result of one modified Newton iteration for the equation

$$hf(y) - k_s = 0, \tag{A.18}$$

using Y_s as starting value. For given k_s this equation can be interpreted as a collocation equation for a numerical solution. Hence, if the property of stiff accuracy holds, if J is invertible and Y_s a sufficiently good starting guess, then the Rosenbrock solution y_{n+1} is close to a collocation solution. Observe that for linear systems $\dot{y} = Jy$ we always have $hJy_{n+1} = k_s$ according to (A.16). If the final increment vector k_s is close to a true derivative, this collocation property seems recommendable. Other arguments supporting the notion of stiff accuracy for nonlinear problems do not exist as far as we know.