Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational and Applied Mathematics 191 (2006) 246–258

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Adaptive stiff solvers at low accuracy and complexity

Alessandra Jannelli, Riccardo Fazio*

Department of Mathematics, University of Messina, Salita Sperone 31, 98166 Messina, Italy

Received 28 January 2005

Abstract

This paper is concerned with adaptive stiff solvers at low accuracy and complexity for systems of ordinary differential equations. The considered stiff solvers are: two second order Rosenbrock methods with low complexity, and the BDF method of the same order. For the adaptive algorithm we propose to use a monitor function defined by comparing a measure of the local variability of the solution times the used step size and the order of magnitude of the solution instead of the classical approach based on some local error estimation. This simple step-size selection procedure is implemented in order to control the behavior of the numerical solution. It is easily used to automatically adjust the step size, as the calculation progresses, until user-specified tolerance bounds for the introduced monitor function are fulfilled. This leads to important advantages in accuracy, efficiency and general ease-of-use. At the end of the paper we present two numerical tests which show the performance of the implementation of the stiff solvers, with the proposed adaptive procedure.

Keywords: Stiff ordinary differential equations; Linearly implicit and implicit numerical methods; Adaptive step size

1. Introduction

The main concern of this work is to study the accuracy, complexity and stability properties of the most promising solvers used for the numerical integration of stiff systems of ordinary differential equations (ODEs) written here, without loss of generality, in autonomous form:

$$\frac{\mathrm{d}c}{\mathrm{d}t} = R(c), \quad t \in [t^0, t_{\max}], \tag{1.1}$$

where $c \in \mathbb{R}^n$ and $R(c) : \mathbb{R}^n \to \mathbb{R}^n$.

Adaptive stiff solvers at low accuracy and complexity are of great interest in the numerical simulation of complex mathematical models as well as in the so-called “real-time” prediction of dangerous situations. Those predictions are within the core development of weather forecasting, nuclear plant engineering, automate control system for vehicles

* Corresponding author.

E-mail addresses: jannelli@dipmat.unime.it (A. Jannelli), rfazio@dipmat.unime.it (R. Fazio)

URL: http://mat520.unime.it/fazio/ (R. Fazio).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.06.041

(airplanes, cars or shuttles), etc. Three-dimensional advection–diffusion–reaction systems are examples of complex mathematical models. One of the simplest numerical approaches for the solution of these models is the so-called operator-splitting where the time evolution of the advection–diffusion part of the system is uncoupled with respect to the reaction part (see, for more details on this topic, the concluding section).

Adaptive ODE solvers can be used to automatically adjust the step size, as the calculation progresses, until a user-specified tolerance is reached. This gives the user control in specifying only the desired tolerance without the need of choosing and changing step value during the calculation.

In the quest for efficient numerical solution of initial value problems (IVPs) governed by ODEs the question of variable step-size selection has been a fundamental one. Accepted strategies for variable step-size selection are based mainly on the inexpensive monitoring of the local truncation error:

(A1) Milne's device in the implementation of predictor-corrector methods;
(A2) embedded Runge–Kutta methods developed by Sarafyan [14], Fehlberg [8], Verner [19] and Dormand and Price [4];
(A3) Richardson local extrapolation [2], as reported by Hairer et al. [9, pp. 228–233].

Moreover, different viewpoints have been also considered in the specialized literature. Promising approaches are listed below:

(B1) residual (or size of the defect) monitoring, proposed by Enright [5], see also his survey paper [6];
(B2) monitoring the relative change in the numerical solution as discussed by Shampine and Witt [16]. This usually leads to significant advantages in accuracy, efficiency and general ease-of-use.

The simple and inexpensive approach to the adaptive step-size selection, considered here, is to require that the change in the solution is monitored in order to define a suitable local step size. This leads to a simpler algorithm than the classical approaches as shown in detail in Sections 3–4.

The paper is organized as follows. In the next section, we focus attention on two second order solvers that have been found particularly useful for dealing with stiff IVPs. In fact, the considered methods are both A- and L-stable. The central core of the paper is Section 3, where we define an adaptive step-size selection procedure and explain the meaning of the related monitor function. Two test problems are used in Section 4 to assess the accuracy of the algorithms resulting from the interplay of the numerical methods and the adaptive procedure. The last section is devoted to discuss future direction of research within the general topic of three-dimensional advection–diffusion–reaction models.

2. The stiff solvers

Stiffness in the numerical solution of IVPs for ODEs of type (1.1) occurs when the system of differential equations involves two or more very different scales of the independent variable on which the dependent variables are changing. Explicit numerical methods are unsuitable to be used in this case, because too small time steps and too long calculation times are necessary to resolve the solution variation on the smallest scale. Hence, the universal choice for stiff problems is to apply implicit methods.

In this section we consider stiff solvers for the numerical approximation of the solutions of the autonomous system (1.1). In particular, the considered methods may be applied to advection–diffusion–reaction models, so that the following criteria should be fulfilled by the solvers:

low accuracy, because of the uncertainty of the available data;
low complexity, due to the huge complexity of the considered problems;
A- and L-stability, that is, stability for large step sizes, and the capacity to follow whatsoever fast transient behavior of the solution;
positivity, meaning that positive solutions should be approximated by positive values;
in the case of mass balance models, mass conservation must be preserved in the computational domain.

2.1. The Rosenbrock methods

In general, the numerical methods for stiff problems use some implicit discretization formulae for reasons of numerical stability. As a consequence, we have to solve a nonlinear system and to this end, the most reliable approach is to apply Newton's method, which demands that the user specifies the Jacobian matrix which is evaluated at each iteration. One possibility to get low complexity for a stiff solver is to avoid these iterations. This is the simple idea for introducing the Rosenbrock methods, which implement the Jacobian matrix directly into the numerical formula rather than within the iterations of Newton's method. To explain how this is achieved, let us consider the simple implicit Euler method

$$c^{n+1} = c^n + \Delta t^n R(c^{n+1}), \tag{2.1}$$

where $\Delta t^n = t^{n+1} - t^n$ for all integer n, and apply only one iteration of Newton's method, so that

$$c^{n+1} = c^n + k, \quad k = \Delta t^n R(c^n) + \Delta t^n Jk, \tag{2.2}$$

where J is the Jacobian matrix given by the derivatives of vector function R with respect to c evaluated at time t^n. In (2.2) we have to solve a linear system of algebraic equations with mass matrix $I - \Delta t^n J$ (here and in the following I is the identity matrix of order n), in which an increment function k appears, rather than a system of nonlinear equations as in (2.1). However, (2.2) retains the A- and L-stability of the implicit Euler method.

In general, s-stage Rosenbrock methods have the following form [10, p. 111]:

$$c^{n+1} = c^n + \sum_{i=1}^{s} b_i k_i,$$

$$k_i = \Delta t^n R\left(c^n + \sum_{j=1}^{i-1} \alpha_{ij} k_j\right) + \Delta t^n J \sum_{j=1}^{i} \delta_{ij} k_j,$$

where s and the coefficients b_i, α_{ij} and δ_{ij} are chosen to obtain a desired order of consistency and suitable stability properties. In particular, we consider the methods where $\delta_{ii} = \delta$ for $i = 1, 2, \ldots, s$; this implies that s linear systems with the same mass matrix, that is $I - \delta \Delta t^n J$, have to be solved at each integration step of the s-stage method, and results in a very low complexity because we can use the same matrix factorization for each of the k_i. For this reason, Rosenbrock methods are called linearly implicit. When $s = 1$, we obtain the above linearized implicit Euler formula, for $s = 2$ we have an infinity of second order Rosenbrock methods

$$\begin{aligned} c^{n+1} &= c^n + b_1 k_1 + b_2 k_2, \\ k_1 &= \Delta t^n R(c^n) + \Delta t^n \delta J k_1, \\ k_2 &= \Delta t^n R(c^n + \alpha_{21} k_1) + \Delta t^n \delta_{21} J k_1 + \Delta t^n \delta J k_2, \end{aligned} \tag{2.3}$$

which have to verify the order conditions

$$b_1 + b_2 = 1, \quad b_2(\alpha_{21} + \delta_{21}) = \tfrac{1}{2} - \delta.$$

Moreover, if we apply (2.3) to the scalar stability test problem $c' = \lambda c$, it provides the numerical solution

$$c^{n+1} = p(z) c^n, \quad p(z) = \frac{1 + (1 - 2\delta) z + (\delta^2 - 2\delta + 1/2) z^2}{(1 - \delta z)^2}, \quad z = \Delta t^n \lambda,$$

so that the above methods are A-stable for $\delta \geqslant \frac{1}{4}$ and L-stable, that is $\lim_{z \to \infty} p(z) = 0$, if and only if $\delta = 1 \pm \frac{1}{\sqrt{2}}$. Numerical tests performed by Verwer et al. [21] pointed out that the value of $\delta = 1 + \frac{1}{\sqrt{2}}$ is the more convenient from a stability viewpoint, so that we will use only this value. Here, and in the following, the $'$ notation indicates differentiation with respect to t.

The first method, we would like to consider, is the one chosen by Verwer et al. [21] and called ROS2:

$$\begin{aligned}
&c^{n+1} = c^n + \tfrac{1}{2}(k_1 + k_2),\\
&k_1 = \Delta t^n R(c^n) + \Delta t^n \delta J k_1,\\
&k_2 = \Delta t^n R(c^n + k_1) - 2\Delta t^n \delta J k_1 + \Delta t^n \delta J k_2
\end{aligned} \tag{2.4}$$

obtained by the chosen parameters

$$b_1 = 1 - b_2, \quad \alpha_{21} = 1/(2b_2), \quad \delta_{21} = -\delta/b_2 \quad \text{with } b_2 = \tfrac{1}{2}.$$

Keeping in mind the low complexity requirement, a different second order Rosenbrock method was chosen by Shampine and Reichelt [15]:

$$\begin{aligned}
&c^{n+1} = c^n + k_2,\\
&k_1 = \Delta t^n R(c^n) + \Delta t^n \delta J k_1,\\
&k_2 = \Delta t^n R(c^n + \tfrac{1}{2}k_1) - \Delta t^n \delta J k_1 + \Delta t^n \delta J k_2,
\end{aligned} \tag{2.5}$$

corresponding to the following parameters:

$$b_1 = 0, \quad b_2 = 1, \quad \alpha_{21} = \tfrac{1}{2}, \quad \delta_{21} = -\delta,$$

which has a very low complexity. This method will be denoted henceforth by ROSE2. We decided to implement both methods because ROS2 represents the method used within huge computations for air pollution models and ROSE2 the state-of-the-art method.

2.2. The BDF method

BDF (backward difference formulas) methods, introduced by Curtiss and Hirschfelder [3], are multistep methods suitable to cope with stiff problems and fast transient behavior of solutions [9, pp. 350–352]. They represent a class of implicit linear multi-step methods with regions of absolute stability large enough to make them relevant to the problem of stiffness.

In this paper, we are interested on methods at low order of accuracy. Then, we consider the second order BDF method with variable time steps (BDF2V)

$$\kappa_0 c^{n+1} + \kappa_1 c^n + \kappa_2 c^{n-1} = R(c^{n+1}), \tag{2.6}$$

where

$$\kappa_0 = \frac{2\Delta t^n + \Delta t^{n-1}}{\Delta t^n(\Delta t^n + \Delta t^{n-1})}, \quad \kappa_1 = -\frac{\Delta t^n + \Delta t^{n-1}}{\Delta t^{n-1}\Delta t^n}, \quad \kappa_2 = \frac{\Delta t^n}{\Delta t^{n-1}(\Delta t^n + \Delta t^{n-1})},$$

which generalizes the A- and L-stable second order BDF method with fixed step size.

The BDF2V method is a two-step one, hence a one-step method has to be applied to the first time step. Assigning the value c^0 at initial time t^0, in order to obtain the value of c^1 at time $t^1 = t^0 + \Delta t^0$, we used the implicit Euler's method applied with iterations of Newton's method. In this way, the order of consistence and the stability of the BDF2V method are left invariant.

3. A simple adaptive step-size strategy

In this section, we present a simple adaptive procedure for determining the local integration step size according to user-specified criteria. Given a step size Δt^n and an initial value c^n at time t^n, the method computes an approximation c^{n+1} at time $t^{n+1} = t^n + \Delta t^n$. Then we can define the following monitor function:

$$\eta^n = \frac{\|c^{n+1} - c^n\|}{\|c^n\| + \varepsilon_M},$$

where $\varepsilon_M > 0$ is of the order of the rounding unit, so that we can require that the step size is modified as needed in order to keep η^n between chosen tolerance bounds, say $0 < \eta_{\min} \leqslant \eta^n \leqslant \eta_{\max}$. The basic guidelines for setting the step size are given by the following algorithm:

(1) Given a step size Δt^n and an initial value c^n at time t^n, the method computes a value c^{n+1} and, consequently, a monitor function η^n by the above formula.
(2) If $\eta_{\min} \leqslant \eta^n \leqslant \eta_{\max}$, then t^n is replaced by $t^n + \Delta t^n$; the step size Δt^n is not changed and the next step, subject to the check at Point (6), is taken by repeating Point (1) with initial value c^n replaced by c^{n+1}.
(3) If $\eta^n < \eta_{\min}$, then t^n is replaced by $t^n + \Delta t^n$ and Δt^n is replaced by $\rho\Delta t^n$ where $\rho > 1$ is a step-size amplification factor; the next integration step, subject to the checks at Points (5) and (6), is taken by repeating Point (1) with initial value c^n replaced by c^{n+1}.
(4) If $\eta^n > \eta_{\max}$, then t^n remains unchanged; Δt^n is replaced by $\sigma\Delta t^n$, where $0 < \sigma < 1$ and the next integration step, subject to the check at Point (5), is taken by repeating Point (1) with the same initial value c^n.
(5) If $\Delta t_{\min} \leqslant \Delta t^n \leqslant \Delta t_{\max}$, return to Point (1); otherwise Δt^n is replaced by $\Delta t_{\max}$ if $\Delta t^n > \Delta t_{\max}$ or by $\Delta t_{\min}$ if $\Delta t^n < \Delta t_{\min}$, then proceed with Point (1).
(6) If $t^n > t_{\max}$, then we set $t^n = t_{\max}$, and $\Delta t^n = t_{\max} - t^{n-1}$.

So that the user has to define the following values:

Δt^0 the initial step size;
$\Delta t_{\min}$, $\Delta t_{\max}$ minimum and maximum values of the step that can be used;
ρ step amplification factor;
σ step reduction factor;
$\eta_{\min}$, $\eta_{\max}$ lower and upper bounds for the tolerance.

A crucial point for any adaptive approach is that the user must set a suitable initial time step. However, with our approach this is not an issue: cf. the results reported in the next section where we used $\Delta t^0 = 0.5\Delta t_{\max}$. Moreover, large enough tolerance intervals for Δt^n and η^n should be used, so that the adaptive procedure does not get caught in a loop, trying repeatedly to modify the step size at the same point in order to meet the bounds that are too restrictive for the given problem. Note that, in general, the step size should not be too small because the number of steps will be large, leading to increased round-off error and computational inefficiency. On the other hand, $\Delta t_{\max}$ should not be too large because local truncation error will be large in this case.

In order to explain the meaning of the monitor function we recall the definition of η^n:

$$\eta^n = \frac{\|c^{n+1} - c^n\|}{\|c^n\| + \varepsilon_M},$$

and note that this can be considered as a measure of the suitability of the used step size to deal with the considered IVP. In fact, we can write

$$\eta^n = \frac{\|c^{n+1} - c^n\|}{\|c^n\| + \varepsilon_M} = \frac{\|c^{n+1} - c^n\|}{\Delta t^n}\,\frac{\Delta t^n}{\|c^n\| + \varepsilon_M} \cong \left\|\frac{\mathrm{d}c}{\mathrm{d}t}(t^n)\right\|\frac{\Delta t^n}{\|c^n\| + \varepsilon_M},$$

where we consider

$$\left\|\frac{\mathrm{d}c}{\mathrm{d}t}(t^n)\right\|$$

as a measure of the increase or decrease of the solution, Δt^n as the grid resolution, and $\|c^n\| + \varepsilon_M$ the order of magnitude of the solution, so that in the above formula the derivative times grid resolution is compared with the order of magnitude of the solution. When the numerical solution increases or decreases too much, our algorithm chooses to reduce the time step. On the other hand, if the solution slowly varies with respect to the grid resolution, then the step size is unchanged or magnified.

We note that our adaptive approach is based on implementation of only one numerical method (of second order ROS2, ROSE2 or BDF2V in our case) that, in order to advance the computation, uses two numerical approximations, obtained at two following time steps.

4. Test problems and numerical results

In this section we consider two test problems and the related numerical results. For both tests the two Rosenbrock methods gave very similar numerical results, so that in the following we shall present only graphical representation of those obtained by the ROS2 method.

4.1. A scalar flame propagation problem

We consider first a simple scalar case within the field of flame propagation [13]

$$c' = c^2(1-c), \quad t \in [0, 2 \cdot 10^4], \tag{4.1}$$

with initial condition

$$c(0) = 10^{-4}. \tag{4.2}$$

This problem has a transient at the middle point of the interval of integration: in fact, the solution changes from being no-stiff to stiff, to become no-stiff again afterwards. For $c \ll 1$ the governing equation is practically $c' = c^2$ whose solution $c(t) = 1/(10^4 - t)$ becomes infinite at $t = 10^4$; but the factor $1 - c$, on the right-hand side, does not allow the solution to become greater than 1. So that at $t = 10^4$ we got a transition from $c \ll 1$ to $c \approx 1$.

This problem is reported here because of the erratic behavior of the Rosenbrock methods with high values of the tolerance $\eta_{\max}$. Figs. 1–3 show the numerical results obtained with our adaptive step-size algorithm. These figures show the numerical solution in the top frame, the step-size selection in the middle frame, and the monitor function in the

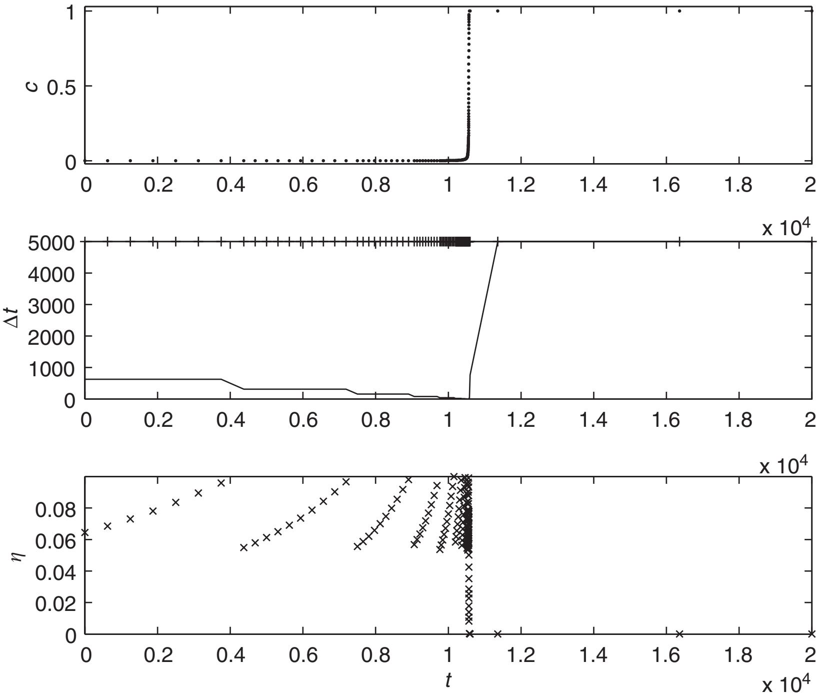

Fig. 1. The flame propagation problem solved by the ROS2 method. Here, we set $\eta_{\max} = 10^{-1}$.

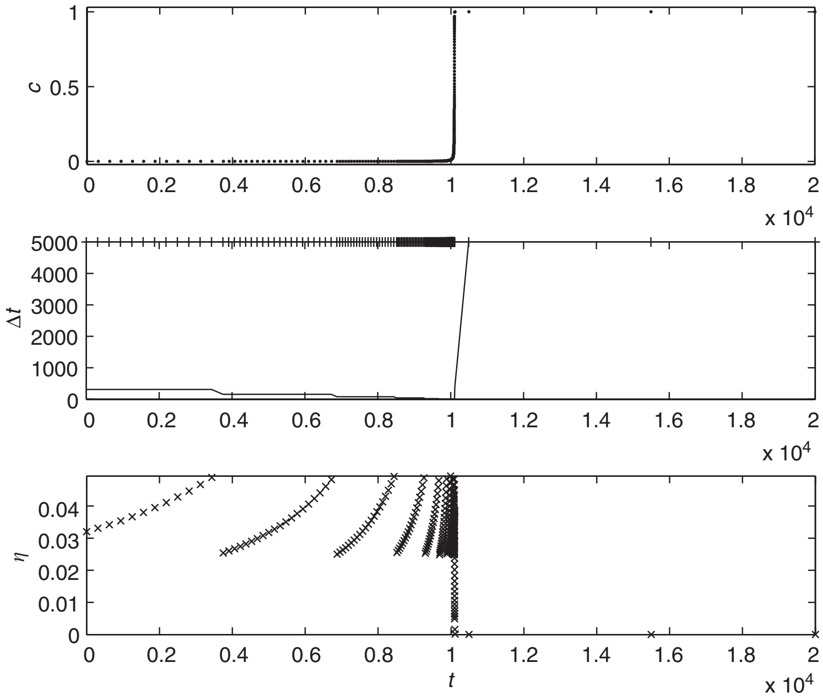

Fig. 2. The flame propagation problem solved by the ROS2 method. In this case, we used $\eta_{\max} = 5 \cdot 10^{-2}$.

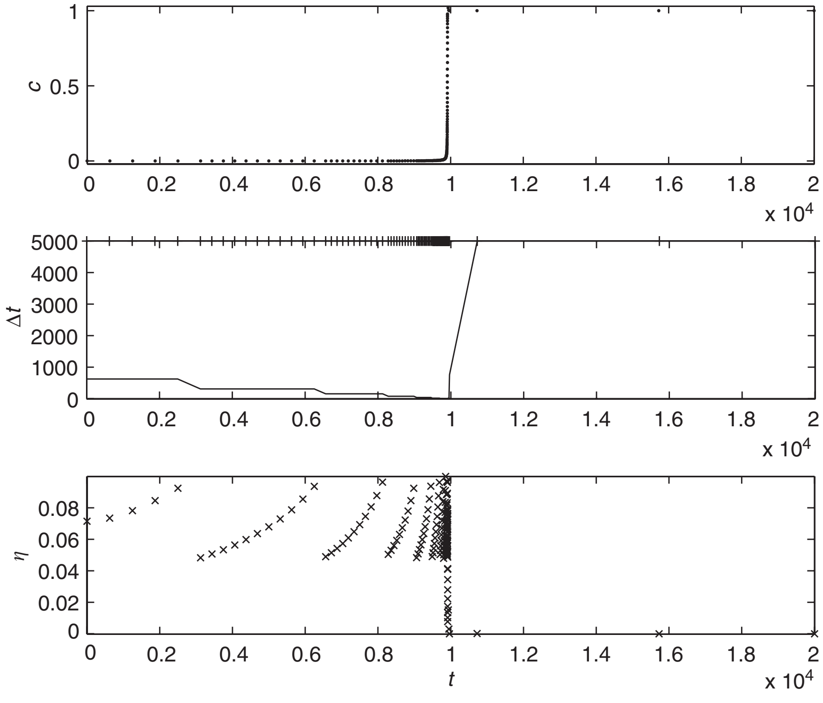

Fig. 3. The flame propagation problem solved by the BDF2V method with $\eta_{\max} = 10^{-1}$.

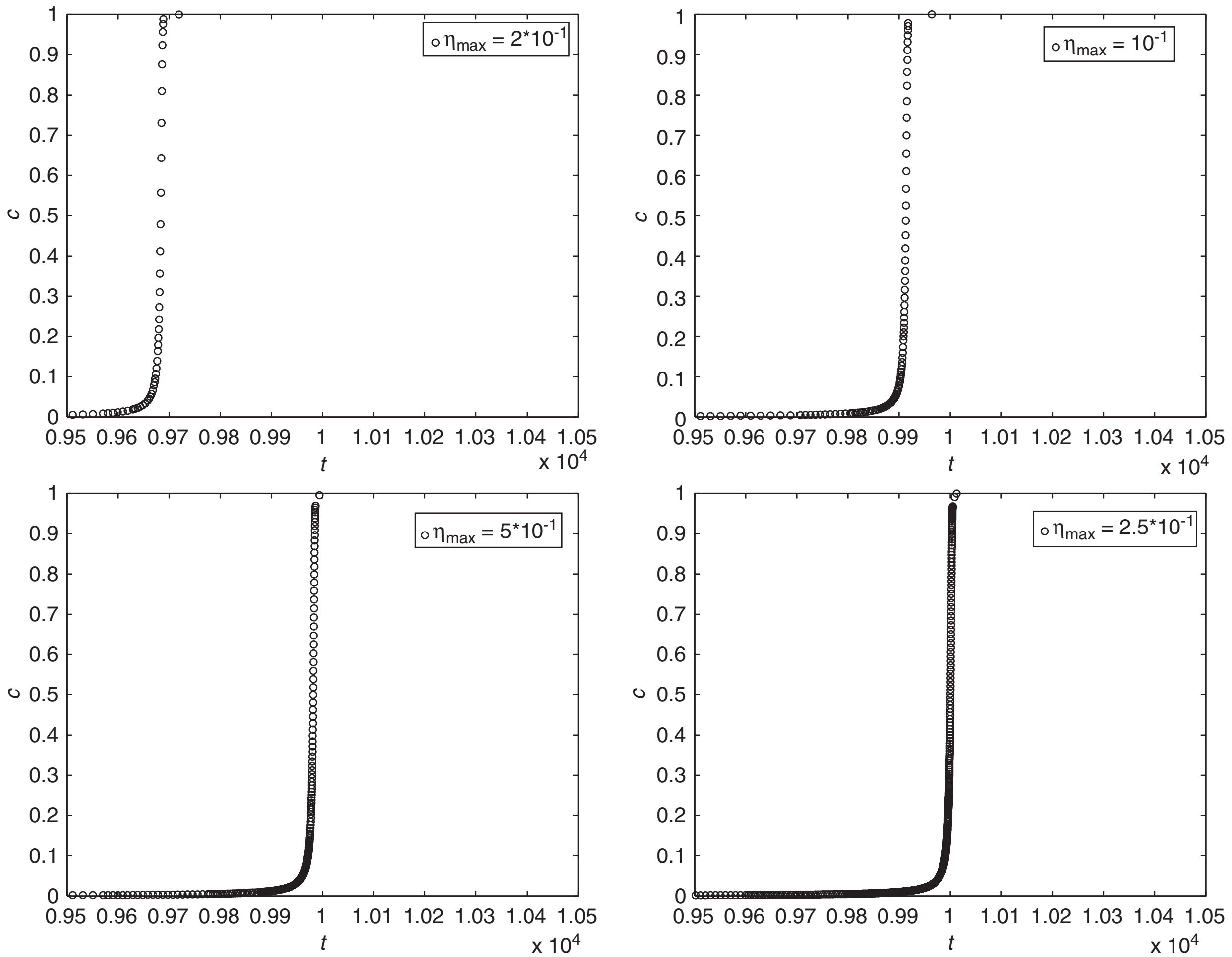

Fig. 4. BDF2V method: detail on the transition point. Top-left: $\eta_{\max} = 2 \cdot 10^{-1}$. Top-right: $\eta_{\max} = 10^{-1}$. Bottom-left: $\eta_{\max} = 5 \cdot 10^{-2}$. Bottom-right: $\eta_{\max} = 2.5 \cdot 10^{-2}$.

bottom frame. It is easily seen in Fig. 1 that, for large tolerances, the transition point is miscalculated by Rosenbrock methods. Even with $\eta_{\max} = 5 \cdot 10^{-2}$ the transition point is calculated after $t = 1$, as shown by Fig. 2.

As far as the adaptive parameters are concerned, we used $\Delta t_{\min} = 0.0005$, $\Delta t_{\max} = 5000$, $\rho = 50$, $\sigma = 0.5$, and $\eta_{\min} = 0.1 \cdot \eta_{\max}$, for all methods. In order to obtain the solution shown in Fig. 1, the algorithm requires 141 steps for ROS2 method plus 13 rejected steps. Moreover, the method uses a minimum step size $\min_n\{\Delta t^n\} = 0.3052$. The solution shown in Fig. 2 was obtained after 285 steps by ROS2 plus 14 rejected steps and $\min_n\{\Delta t^n\} = 0.1526$. On the other hand, 150 steps, plus 13 rejected, were used by the BDF2V method to compute the results shown in Fig. 3.

A comparison can be made with the steps used by the two stiff methods implemented within the MATLAB ODE suite by Shampine and Reichelt [15]. As reported by Moler [12], by using a relative error tolerance of 10^{-3}, the `ode23s` routine used only 55 steps and the `ode15s` one 103 steps; but we do not know how many rejected steps were used in those cases. As it is easily seen, our step selection strategy is more expensive with respect to the state-of-the-art software but our algorithm is simpler and faster with respect to the algorithm implemented in the MATLAB suite.

We conclude this test by reporting the results displayed in Fig. 4 where we focus our attention on the transition point, proposing a simple accuracy test by comparing the numerical results obtained for three successive halving of the value of $\eta_{\max}$.

4.2. A simple air pollution model

In this test we illustrate the mass action law by three reactions between oxygen O_2, atomic oxygen O, ozone O_3, nitrogen oxide NO and nitrogen dioxide NO_2. It is important, first, to consider the primary chemical reactions that are involved:

$$NO_2 + hv \overset{\mu_1(t)}{\rightarrow} NO + O$$

$$O + O_2 \overset{\mu_2}{\rightarrow} O_3$$

$$NO + O_3 \overset{\mu_3}{\rightarrow} O_2 + NO_2$$

The first photochemical reaction says that during the light hours, due to the solar radiation indicated here by hv, NO_2 is photo-dissociated into NO and O; this reaction is regulated by $\mu_1(t)$ as specified below. We assume that the oxygen concentration O_2 is constant, which is a realistic hypothesis.

Setting the concentrations $c_1 = [O]$, $c_2 = [NO]$, $c_3 = [NO_2]$ and $c_4 = [O_3]$, a simple model used by Verwer et al. [20] for the air pollution in the lower troposphere is given below:

$$\begin{aligned} c_1' &= \mu_1(t)c_3 - \mu_2 c_1, \\ c_2' &= \mu_1(t)c_3 - \mu_3 c_2 c_4 + s_2, \\ c_3' &= \mu_3 c_2 c_4 - \mu_1(t)c_3, \\ c_4' &= \mu_2 c_1 - \mu_3 c_2 c_4. \end{aligned} \tag{4.3}$$

The concentrations are given in molecules for cm^3 and time in seconds. Note that μ_2 is the total number of oxygen molecules per cm^3 and as a consequence is much larger than $\mu_1(t)$ and μ_3. Moreover, a constant source term s_2 is used to simulate the emission of nitrogen oxide.

The reported numerical results use the initial conditions

$$c(t^0) = [0, 1.3 \cdot 10^8, 5 \cdot 10^{11}, 8 \cdot 10^{11}]^T,$$

and the following involved parameters:

$$\mu_1(t) = \begin{cases} 10^{-40} & \text{night-hours: 8 p.m.–4 a.m.,} \\ 10^{-5}\, e^{7 \sec(t)} & \text{day-hours: 4 a.m.–8 p.m.,} \end{cases}$$

$$\mu_2 = 10^5, \quad \mu_3 = 10^{-16}, \quad s_2 = 10^6,$$

where

$$\sec(t) = \left(\sin\left(\frac{\pi}{16}(t_h - 4)\right)\right)^{0.2}$$

and

$$t_h = th - 24\lfloor th/24 \rfloor, \quad th = t/3600,$$

here $\lfloor z \rfloor$ stands for the floor function.

Fig. 5 shows the results for a period of five days, that is, from 4 a.m. ($t^0 = 14\,400$) up to 4 a.m. of the next five days ($t_{\max} = 504\,000$), obtained with ROS2, ROSE2 and BDF2V methods.

For the adaptive algorithm we set $\eta_{\max} = 10^{-3}$ and $\eta_{\min} = 0.1 \cdot \eta_{\max}$, $\rho = 50$, $\sigma = 0.5$, $\Delta t_{\max} = 1000$ and $\Delta t_{\min} = .1$. At this value of $\eta_{\max}$ the three methods compute the same solution within the used graphical scales. The BDF2V method requires 21 255, plus 102 rejected, steps; whereas the ROS2 uses 21 343 steps plus 106 rejection. Both methods apply the maximum allowed step size $\Delta t_{\max}$ and $\min_n\{\Delta t^n\} = 1.5259$.

Figs. 6 and 7 show the step-size selection and the monitor function for the ROS2 and BDF2V methods, respectively.

This problem is considered because we would like to show how our adaptive approach is able to deal with a system and how the considered numerical methods preserve positivity, see the solution's first component in Fig. 5, and mass conservation.

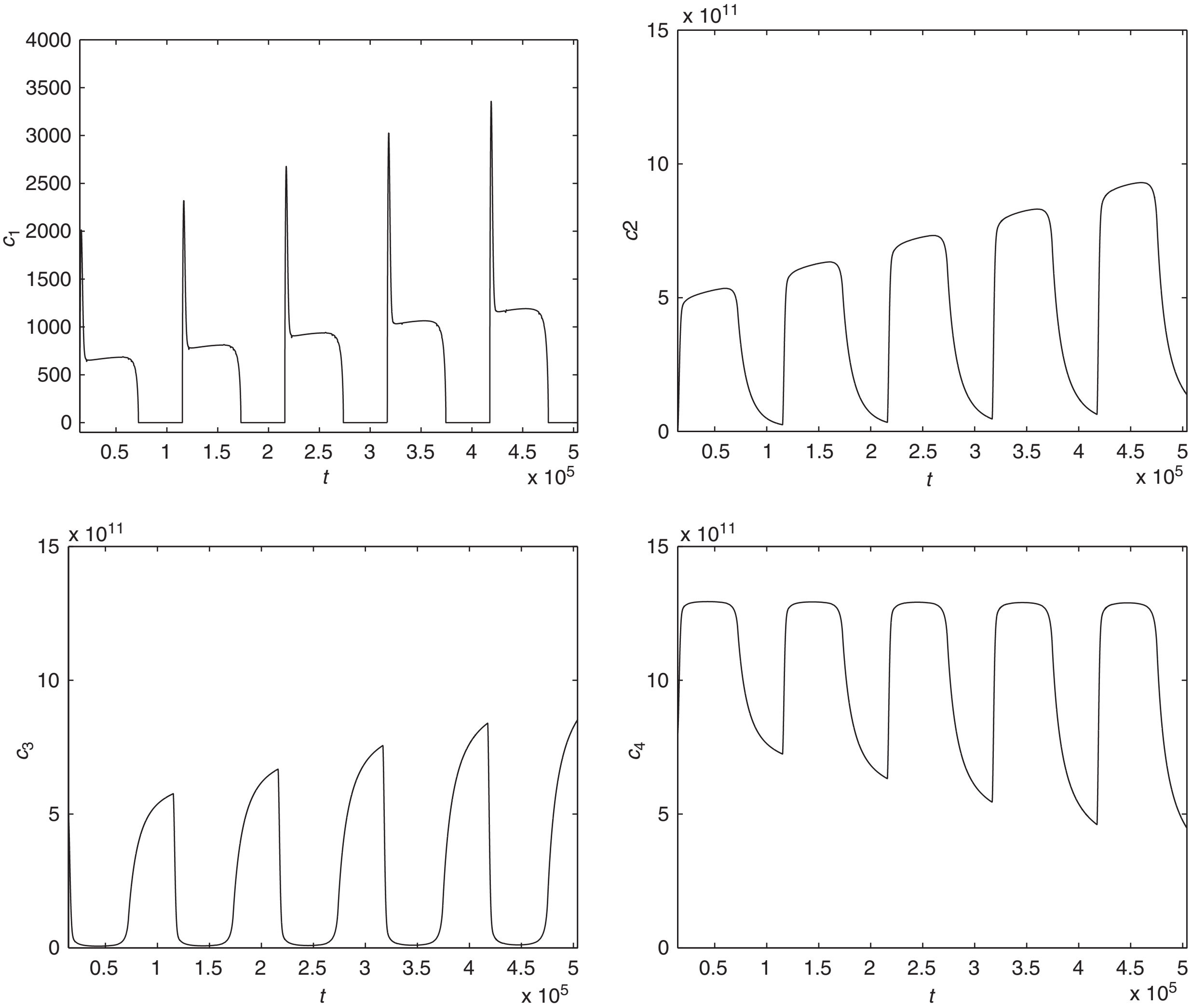

Fig. 5. The air pollution model: numerical solution obtained by ROS2, and BDF2V.

Molecular conservation mass law associated with $R(c)$ in system (4.3)

$$\sum_{i=1}^{4} \beta_i R_i(c) = 0 \quad \text{with} \quad \beta_i \geqslant 0, \quad i = 1, \ldots, 4$$

implies that for every time, the mass of the associated set of species is conserved:

$$\sum_{i=1}^{4} \beta_i c_i(t) = \text{constant}.$$

Two mass laws for this chemical model exist:

$$c_1'(t) + c_3'(t) + c_4'(t) = 0 \quad \text{and} \quad c_2'(t) + c_3'(t) = s_2,$$

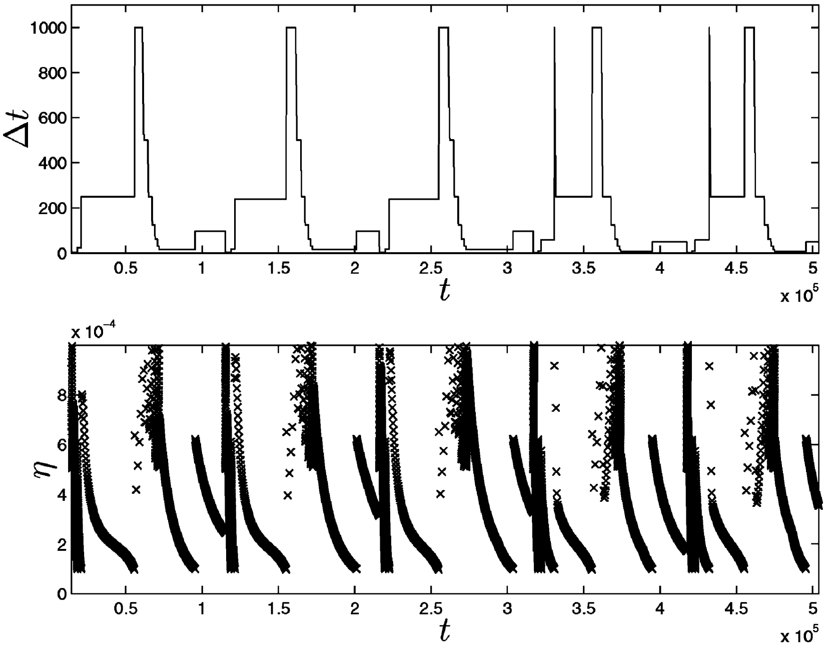

Fig. 6. The air pollution model: ROS2 step-size selection (top) and monitor function (bottom).

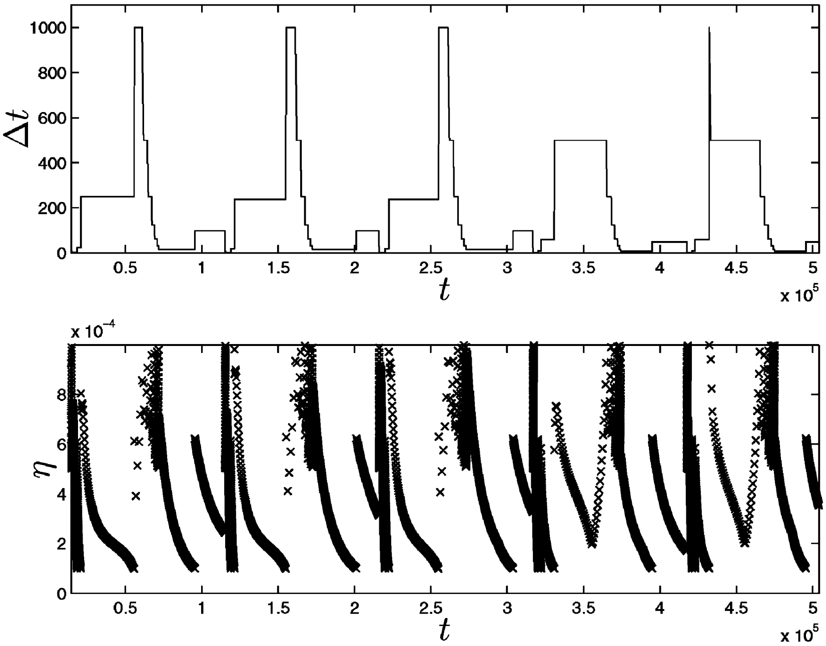

Fig. 7. The air pollution model: BDF2V step-size selection (top) and monitor function (bottom).

hence $[\mathrm{O}] + [\mathrm{NO}_2] + [\mathrm{O}_3]$ is a conserved quantity while $[\mathrm{NO}] + [\mathrm{NO}_2]$ grows with $s_2 t$. The first mass law shows that the quantity $c_1(t) + c_3(t) + c_4(t)$ is constant in time, that is,

$$c_1(t) + c_3(t) + c_4(t) = c_1(t^0) + c_3(t^0) + c_4(t^0) = 1.3\mathrm{E} + 12,$$

Table 1
One day relative error computed for the second mass law

Time	Exact solution	ROS2	ROSE2	BDF2V
04 a.m.	5.0013E + 11			
12 a.m.	5.2906E + 11	1.0383E − 15	1.9612E − 15	2.3188E − 14
08 p.m.	5.5773E + 11	8.7548E − 16	2.4076E − 15	2.9548E − 14
04 a.m.	5.8653E + 11	1.5384E − 04	1.5384E − 04	2.0646E − 05

then the numerical solution should verify this condition every time. Indeed, all the considered methods (ROS2, ROSE2 and BDF2V) satisfy the above condition within a relative error smaller than 1E − 12. By the second mass law the quantity $c_2(t) + c_3(t)$ grows with $s_2 t$, then we have

$$c_2(t) + c_3(t) = s_2 t + c_2(t^0) + c_3(t^0) - s_2 t^0.$$

Considering the numerical solutions and the exact relation reported above, we can calculate the relative error showed in Table 1. It is easily seen that the considered stiff solvers with our adaptive procedure provide accurate conservation of these mass laws.

5. Conclusions

In the introduction we reported the background for the proposed adaptive step-size strategy. Phenomena of relevant interest, such as air pollution [20] or marine pollution [18], prey–predator evolution in ecosystems [11], etc., can be studied by using three-dimensional advection–diffusion–reaction models. Let us consider, as an example of complex models, a three-dimensional advection–diffusion–reaction model governed by the following system of equations:

$$\frac{\partial \mathbf{c}}{\partial t} + \nabla \cdot (\mathbf{v}\mathbf{c}) - \nabla \cdot (D \nabla \mathbf{c}) = R(\mathbf{c}), \tag{5.1}$$

where $\mathbf{c} = \mathbf{c}(\mathbf{x}, t)$ with $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{x} \in \Omega \subset \mathbb{R}^3$, t and $\mathbf{x}$ denote time and space variables, respectively; the advection field $\mathbf{v}$ and the diffusion coefficient matrix D are, usually, supposed to be given. Let us stress that the left-hand side of (5.1) has a scalar nature, that is, each component of the field variable $\mathbf{c}$ is governed by a scalar partial differential equation, in contrast with its right-hand side where all the components are coupled. Moreover, the time evolution of each component of the left-hand side is determined by a partial differential equation, whereas the time evolution of the right-hand side has a local (in space) dependence. For the above reasons one of the simplest numerical approaches for the solution of (5.1) is the so-called operator-splitting [17] where the time evolution of the advection–diffusion part of the system is uncoupled with respect to the reaction part.

Our main topic, that is, the adaptive implementation of numerical methods, is a fundamental one also for the numerical solution of partial differential problems. Indeed, the superiority of numerical schemes using moving mesh methods against schemes with constant step sizes has been proved for several classes of problems: see Budd and Collins [1] for parabolic problems with blowing up solutions and Fazio and LeVeque [7] for hyperbolic conservation laws.

Acknowledgements

The research of this work was supported by a grant of the Messina University, by INDAM 2003 project “Modellistica Numerica per il Calcolo Scientifico ed Applicazioni Avanzate” and partially by the Italian “MIUR”.

References

[1] C.J. Budd, G.J. Collins, Symmetry based numerical methods for partial differential equations, in: D.F. Griffiths, D.J. Higham, G.A. Watson (Eds.), Numerical Analysis 1997, Pitman Research Notes in Mathematics, vol. 380, Longman, Harlow, 1998, pp. 16–36.

[2] R. Bulirsch, J. Stoer, Numerical treatment of ordinary differential equations by extrapolation methods, Numer. Math. 8 (1966) 1–13.
[3] C.F. Curtiss, J.O. Hirschfelder, Integration of stiff equations, Proc. Natl. Acad. Sci. USA 38 (1952) 235–243.
[4] J.R. Dormand, P.J. Price, A family of embedded Runge–Kutta formulae, J. Comput. Appl. Math. 6 (1980) 19–26.
[5] W.H. Enright, A new error-control for initial value solvers, Appl. Math. Comput. 31 (1989) 588–599.
[6] W.H. Enright, Continuous numerical methods for ODEs with defect control, J. Comput. Appl. Math. 125 (2000) 159–170.
[7] R. Fazio, R.J. LeVeque, Moving-mesh methods for one-dimensional hyperbolic problems using CLAWPACK, Comput. Math. Appl. 45 (2003) 273–398.
[8] E. Fehlberg, Classical fifth-, sixth-, seventh- and eighth order formulas with step size control, Computing 4 (1969) 93–106.
[9] E. Hairer, S. Nørsett, G. Wanner, Solving Ordinary Differential Equations I. Nonstiff Problems, Springer, Berlin, 1987.
[10] E. Hairer, G. Wanner, Solving Differential Equations II. Stiff and Differential–Algebraic Problems, Springer, Berlin, 1991.
[11] A. Jannelli, R. Fazio, D. Ambrosi, A 3D mathematical model for the prediction of mucilage dynamics, Comput. Fluids 32 (2003) 47–57.
[12] C. Moler, Golden ODEs, MATLAB News & Notes, Summer 1996, available on Internet as the file Sum96.Cleve.pdf at the URL: ⟨www.mathworks.com/company/newsletters/news_notes/clevercorner⟩.
[13] R.E. O'Malley, Singular Perturbation Methods for Ordinary Differential Equations, Springer, Berlin, 1991.
[14] D. Sarafyan, Error estimation for Runge–Kutta methods through pseudoiterative formulas, Technical Report No. 14, Lousiana State University, New Orleans, 1966.
[15] L.F. Shampine, M.W. Reichelt, The MATLAB ODE suite, SIAM J. Sci. Comput. 18 (1997) 1–22.
[16] L.F. Shampine, A. Witt, A simple step size selection algorithm for ODEs codes, J. Comput. Appl. Math. 58 (1995) 345–354.
[17] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.
[18] M. Toro, L.C. van Rijn, K. Meijer, Three-dimensional modelling of sand and mud transport in current and waves, Technical Report No. H461, Delft Hydraulics, Delft, The Netherlands, 1989.
[19] J.M. Verner, Explicit Runge–Kutta methods with estimates of the local truncation error, SIAM J. Numer. Anal. 15 (1978) 772–790.
[20] J.G. Verwer, W.H. Hundsdorfer, J.G. Blom, Numerical time integration for air pollution models, Sur. Math. Ind. 2 (2002) 107–174.
[21] J.G. Verwer, E.J. Spee, J.G. Blom, W.H. Hundsdorfer, A second-order Rosenbrock method applied to photochemical dispersion problems, SIAM J. Sci. Comput. 20 (1999) 1456–1480.