UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Multirate numerical integration for ordinary differential equations

Savcenco, V.

Publication date
2008

Link to publication

Citation for published version (APA):
Savcenco, V. (2008). *Multirate numerical integration for ordinary differential equations*. [Thesis, externally prepared, Universiteit van Amsterdam].

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:09 Mar 2023

Chapter 4

Construction of high-order multirate Rosenbrock methods for stiff ODEs

Multirate time stepping is a numerical technique for efficiently solving large-scale ordinary differential equations (ODEs) with widely different time scales localized over the components. This technique enables one to use large time steps for slowly varying components, and small steps for rapidly varying ones. Multirate methods found in the literature are normally of low order, one or two. Focusing on stiff ODEs, in this chapter we discuss multirate methods based on the higher-order, stiff Rosenbrock integrators. Special attention is paid to the treatment of the refinement interfaces with regard to the choice of the interpolant and the occurrence of order reduction. For stiff, linear systems containing a stiff source term, we propose modifications for the treatment of the source term which overcome order reduction originating from such terms and which we can implement in our multirate method.

4.1 Introduction

Many practical applications give rise to systems of ordinary differential equations (ODEs) with different time scales which are localized over the components. To solve such systems, multirate time stepping strategies are considered. These strategies integrate the slow components with large time steps and the fast components with small time steps.

Numerous multirate methods were developed for solving stiff systems with different time scales. A multirate method based on a two stage second-order Rosenbrock method together with a self-adjusting multirate time stepping strategy was introduced in Chapter 1. In [3] a scheme based on a third-order Rosenbrock method was considered. However, due to stability constraints, instead of the third-order method the embedded second-order method was used for time stepping. A multirate method for circuit simulation problems based on the backward Euler method was described in [55]. All these schemes are of order

two at most. In this chapter we aim to develop multirate methods of higher order.

We address the main difficulties which arise in the construction of higher-order multirate methods. Special attention is paid to the treatment of the temporal refinement interface. During the refinement step the intermediate time values of the components which are not refined might be needed. Usually these values are not directly available and have to be calculated by interpolation or a dense output formula. Use of low-order interpolation can influence the order of the method, therefore a better interpolation has to be considered.

We construct a multirate method which is based on the fourth-order Rosenbrock method RODAS of Hairer and Wanner [19]. In the numerical experiments the constructed method is compared with the multirate version of the second order Rosenbrock method ROS2 from Chapter 1. From experiments it is seen that the multirate RODAS shows good results and is more robust than the multirate ROS2.

The contents of this chapter is as follows. In Section 4.2 we discuss the main issues of the high-order Rosenbrock methods construction. In Section 4.3 we describe an interpolant which can be used together with a second-order two stage Rosenbrock method ROS2 [27]. Fourth-order Rosenbrock methods are discussed in Section 4.4. Order reduction issues and the modifications for the Rosenbrock methods which help to avoid order reduction are presented in Section 4.5. In Section 4.6 four test problems are solved using a self-adjusting multirate strategy based on a Rosenbrock fourth-order method. The numerical results are compared with the ones obtained with lower-order Rosenbrock methods. Finally, Section 4.7 contains the conclusions.

4.2 Considerations on construction of high-order multirate Rosenbrock methods

We consider a system of ODEs

$$w'(t) = F(t, w(t)), \qquad w(0) = w_0, \tag{4.1}$$

with given initial value $w_0 \in \mathbb{R}^m$ and given function $F : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$. The approximations to the exact ODE solution at the global time levels t_n will be denoted by w_n. The multirate methods in this chapter are based on the approach described in Chapter 1. For a given global time step $\tau = t_n - t_{n-1}$, we first compute a tentative approximation at the time level t_n for all components. For those components for which an error estimator indicates that smaller steps are needed, the computation is redone with halved step size $\frac{1}{2}\tau$. During the refinement stage, values at the intermediate time levels of components which are not refined might be needed. These values can be obtained by extrapolation, interpolation or by use of dense output built in the time integration method. The refinement is recursively continued until an error estimator is below a prescribed tolerance for all components. A schematic example, with components

horizontally and time vertically, is presented in Figure 4.1.

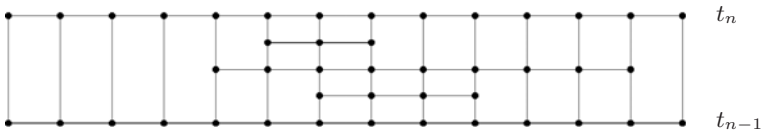

FIGURE 4.1: Multirate time stepping for a time interval $[t_{n-1}, t_n]$.

Proper interface treatment during the refinement step is very important for multirate schemes. Use of interpolation and dense output of order lower than the order of the main time integration method can lead to order reduction. For example, in Chapter 2 it was shown that the second-order trapezoidal rule with linear interpolation can lead to first-order consistency for stiff problems. Another important point in connection with stiff problems, is that interpolation procedures which make explicit use of function evaluations are inappropriate. In this case, the interpolant resulting from a stiff problem can dramatically amplify the error of the numerical method. Such interpolants are usually called "unstable" [4].

Let us consider an s-stage Rosenbrock method [19]

$$w_n = w_{n-1} + \sum_{i=1}^{s} b_i k_i \,, \tag{4.2}$$

$$\begin{aligned} k_i \;=\; & \tau F\left(t_{n-1} + \alpha_i \tau, w_{n-1} + \sum_{j=1}^{i-1} \alpha_{ij} k_j\right) + \tau \frac{\partial F}{\partial w}(t_{n-1}, w_{n-1}) \sum_{j=1}^{i} \gamma_{ij} k_j \\ & + \gamma_i \tau^2 \frac{\partial F}{\partial t}(t_{n-1}, w_{n-1}) \,, \end{aligned} \tag{4.3}$$

where $\alpha_{ij}, \gamma_{ij}, b_i$ are real parameters defining the method, τ denotes the step size, and

$$\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}, \qquad \gamma_i = \sum_{j=1}^{i} \gamma_{ij} \,. \tag{4.4}$$

A dense output or a continuous extension for this method can be defined as

$$w_I(t_{n-1} + \theta\tau) = w_{n-1} + \sum_{i=1}^{s} \theta b_i(\theta) k_i, \qquad 0 \le \theta \le 1 \,. \tag{4.5}$$

In this chapter we mainly consider numerical time integration methods for which there exist interpolants which do not amplify the error of the numerical method within one step for the linear test equation

$$w'(t) = \lambda w(t), \quad w(0) = 1 \,, \tag{4.6}$$

with $\lambda \in \mathbb{C}^-$, where $\mathbb{C}^-$ denotes the left-half complex plane $\{z \in \mathbb{C} : \mathrm{Re}(z) \leq 0\}$. Following the definition presented by Bellen and Zennaro in [4], we will say that the interpolant (4.5) is stable with respect to a Rosenbrock method (4.2)-(4.3) if

$$\max_{0\leq\theta\leq 1} |w_I(\theta\tau)| \leq \max\{1, |w(\tau)|\} \tag{4.7}$$

for every $z = \lambda\tau \in \mathbb{C}^-$.

In case of an A-stable Rosenbrock method, the condition of stability reduces to

$$\max_{0\leq\theta\leq 1} |w_I(\theta\tau)| \leq 1\,, \tag{4.8}$$

for every $z = \lambda\tau \in \mathbb{C}^-$.

An interpolant with this property was considered together with a second-order Rosenbrock method in Chapter 3. A detailed description of this interpolant is given in Section 4.3. This combination resulted in a multirate method which showed good asymptotic stability properties. We believe that an interpolant with property (4.8) will not blow up the error of the associated method, however, the stability analysis of the final multirate scheme is still missing.

For the dense output formula (4.5) used for the Rosenbrock method (4.2)-(4.3), it is possible to derive order conditions, see [19]:

Order 1

$$\sum b_i(\theta) = 1\,, \tag{4.9}$$

Order 2

$$\sum b_i(\theta)\beta_i = \frac{1}{2}\theta - \gamma\,, \tag{4.10}$$

Order 3

$$\sum b_i(\theta)\alpha_i^2 = \frac{1}{3}\theta^2\,, \tag{4.11}$$

$$\sum b_i(\theta)\beta_{ij}\beta_j = \frac{1}{6}\theta^2 - \gamma\theta + \gamma^2\,, \tag{4.12}$$

Order 4

$$\sum b_i(\theta)\alpha_i^3 = \frac{1}{4}\theta^3\,, \tag{4.13}$$

$$\sum b_i(\theta)\alpha_i\alpha_{ik}\beta_k = \frac{1}{8}\theta^3 - \frac{1}{3}\gamma\theta^2\,, \tag{4.14}$$

$$\sum b_i(\theta)\beta_{ik}\alpha_k^2 = \frac{1}{12}\theta^3 - \frac{1}{3}\gamma\theta^2\,, \tag{4.15}$$

$$\sum b_i(\theta)\beta_{ik}\beta_{kl}\beta_l = \frac{1}{24}\theta^3 - \frac{1}{2}\gamma\theta^2 + \frac{3}{2}\gamma^2\theta - \gamma^3\,, \tag{4.16}$$

where

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

Sometimes, for a given Rosenbrock method, it is impossible to define a continuous interpolant (for any $0 \leq \theta \leq 1$). Instead, the discrete version of the

interpolation can be considered, in which the stability and order conditions are satisfied just for few values of the parameter θ. In the case of our multirate time stepping strategy, at each refinement step we have to interpolate time points at the stages. Specifically, for the refinement step where we take two smaller time steps of size $\frac{1}{2}\tau$ instead of one of size τ, we need a stable interpolant for $\theta = \frac{1}{2}(l + \alpha_i)$ for $l = 0, 1$ and $i = 1, ..., s$.

4.3 A stable interpolant for multirate ROS2

In this section we will consider the two-stage second-order Rosenbrock ROS2 method [27]. To proceed from t_{n-1} to a new time level $t_n = t_{n-1} + \tau$, the method calculates

$$\begin{aligned} w_n &= w_{n-1} + \tfrac{3}{2}\bar{k}_1 + \tfrac{1}{2}\bar{k}_2\,, \\ \big(I - \gamma\tau J\big)\bar{k}_1 &= \tau F(t_{n-1}, w_{n-1}) + \gamma\tau^2 F_t(t_{n-1}, w_{n-1})\,, \\ \big(I - \gamma\tau J\big)\bar{k}_2 &= \tau F(t_n, w_{n-1} + \bar{k}_1) - \gamma\tau^2 F_t(t_{n-1}, w_{n-1}) - 2\bar{k}_1\,, \end{aligned} \tag{4.17}$$

where $J \approx F_w(t_{n-1}, w_{n-1})$ and the notation $\bar{k}_i$ instead of k_i is used since we have eliminated the matrix-vector product in the second stage. The method is A-stable for $\gamma \geq \frac{1}{4}$ and L-stable if $\gamma = 1 \pm \frac{1}{2}\sqrt{2}$. We use $\gamma = 1 - \frac{1}{2}\sqrt{2}$. For this method, for $\gamma \neq \frac{1}{2}$, we define the following second-order interpolant

$$w_I(t_{n-1}+\theta\tau) = w_{n-1} + \frac{1}{2(1-2\gamma)}\left(\theta^2 + (2-6\gamma)\theta\right)\bar{k}_1 + \frac{1}{2(1-2\gamma)}\left(\theta^2 - 2\gamma\theta\right)\bar{k}_2\,, \tag{4.18}$$

which was already used in Chapter 3.

For studying the stability of this interpolant we apply it to the test equation (4.6) and use the maximum modulus principle from complex analysis. Thus we have to check whether $\max_{0\leq\theta\leq 1} |w_I(\theta\tau)| \leq 1$ whenever $\mathrm{Re}(z) = 0$, where $z = \lambda\tau$. From Figure 4.2, where the values of $|w_I(\theta\tau)|$ are presented for $\gamma = 1 - \frac{1}{2}\sqrt{2}$ and for three different values of θ, we can see that $|w_I(\theta\tau)|$ does not exceed 1. Experiments also showed that $|w_I(\theta\tau)|$ does not exceed 1 for all $0 \leq \theta \leq 1$ and $\gamma \geq \frac{1}{4}, \gamma \neq \frac{1}{2}$.

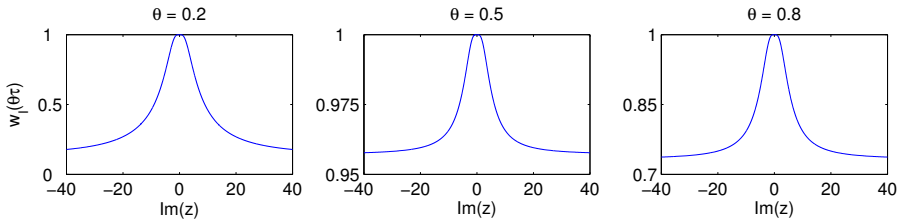

FIGURE 4.2: Plot of $|w_I(\theta\tau)|$ for $\gamma = 1 - \frac{1}{2}\sqrt{2}$ and three different values of θ.

The stability of this interpolant in the sense of definition (4.8) can also be shown analytically. Assuming $w_0 = 1$ and inserting $z = iy$ in (4.17) gives

$$\bar{k}_1 = \frac{iy}{1 - i\gamma y}, \qquad \bar{k}_1 + \bar{k}_2 = \frac{y^2(2\gamma - 1)}{(1 - i\gamma y)^2}\,.$$

The interpolant (4.18) becomes

$$\begin{aligned} w_I(\theta\tau) &= 1 + \theta\bar{k}_1 + \frac{\theta(\theta - 2\gamma)}{2(1 - 2\gamma)}(\bar{k}_1 + \bar{k}_2) \\ &= 1 + \theta\frac{iy}{1 - i\gamma y} - \frac{\theta(\theta - 2\gamma)}{2}\frac{y^2}{(1 - i\gamma y)^2} \\ &= -1 + \frac{1}{2}\theta y\frac{(\theta y - \theta\gamma^2 y^3 + 4\gamma^3 y^3) + (2\theta\gamma y^2 - 6\gamma^2 y^2 - 2)i}{(1 + \gamma^2 y^2)^2} \end{aligned} \tag{4.19}$$

After some simplifications we get

$$|w_I(\theta\tau)|^2 = 1 - \frac{(4\gamma - \theta)(2\gamma - \theta)^2 y^4}{4(1 + \gamma^2 y^2)^4}\,.$$

Since we have $4\gamma - \theta \geq 0$, it follows that $|w_I(\theta\tau| \leq 1$. This shows that the considered interpolant used together with the ROS2 method is stable in the sense of definition (4.8).

4.4 Higher-order multirate methods

In this section we consider some fourth-order Rosenbrock methods well known from the literature: Kaps-Rentrop methods [29] and the RODAS method of Hairer and Wanner [19]. Attempts to construct multirate methods based on the Kaps-Rentrop methods appeared to be not so successful (see Subsection 4.4.2). Therefore the main part of this section is about the multirate version of the RODAS method.

4.4.1 Multirate RODAS

In this subsection we present a multirate method based on the fourth-order stiffly accurate, A-stable Rosenbrock method RODAS [19]. RODAS has six stages and a third-order embedded method which can be used for error estimation. It also has a built-in dense output of order three.

The coefficients of the RODAS method, derived following [19, pp. 421], are presented in Table 4.7 in the Appendix. The embedded method is given by

$$\overline{w}_n = w_{n-1} + \sum_{i=1}^{s} \bar{b}_i k_i\,, \tag{4.20}$$

with $\bar{b}_i = \alpha_{5i}$. The built-in dense output of the RODAS method is defined by

$$w_I(t_{n-1} + \theta\tau) = w_{n-1} + \sum_{i=1}^{s} \sum_{j=0}^{3} b_{ij}\theta^{j+1} k_i \,, \tag{4.21}$$

with the coefficients b_{ij} presented in Table 4.8 in the Appendix. These coefficients were chosen to satisfy the third-order conditions (4.9)-(4.12), the first fourth-order condition (4.13) and the condition $b_6(\theta) = \gamma\theta$, see [19].

In order to test the stability of the dense output in the sense of definition (4.8), we apply the RODAS method together with its dense output to the scalar test equation $w' = \lambda w$. We use the maximum modulus principle and check how the value of $|w_I(\theta\tau)|$ changes for different purely imaginary values of $z = \tau\lambda$. In Figure 4.3 the plot of the $\max|w_I(\theta\tau)|$ for a range of z-values is presented. We can see that the maximum of the modulus of the solution is always smaller than 1.04, which is a slightly larger threshold than in definition (4.8). This also holds for larger values of z. Therefore, the RODAS built-in dense output will not amplify dramatically the error of the main numerical method. Moreover, the RODAS formula itself will provide damping due to its L-stability.

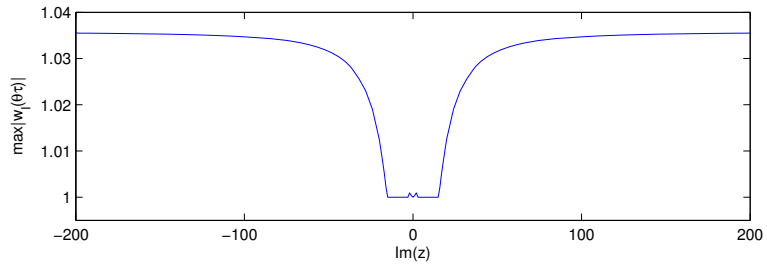

FIGURE 4.3: Plot of the $\max_{0\le\theta\le1}|w_I(\theta\tau)|$ for a range of purely imaginary z-values.

The dense output of RODAS, which is used for interpolation in our multirate scheme, is of order three. Therefore, due to possible order reduction (see [25]), the multirate method based on RODAS is of order three. However, in most practical examples we will see order four due to cancellation and damping.

Asymptotic stability for 2×2 test equations

Usually, linear stability analysis of an integration method is based on the scalar Dahlquist test equation $w'(t) = \lambda w(t)$, $\lambda \in \mathbb{C}$. For multirate methods the scalar problem cannot be used. Instead we can consider a similar test problem, a linear 2×2 system

$$w'(t) = Aw(t), \qquad w = \begin{pmatrix} u \\ v \end{pmatrix}, \qquad A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

We denote

$$Z = \tau A, \qquad z_{ij} = \tau a_{ij} \,. \tag{4.22}$$

We will assume that the first component u of the system is fast and the second component v is slow. Thus, to perform the time integration from t_{n-1} to $t_n = t_{n-1} + \tau$ we will complete two time steps of size $\frac{1}{2}\tau$ for the first component and one time step of size τ for the second component.

We assume that

$$a_{11} < 0 \qquad \text{and} \qquad a_{22} < 0 \,. \tag{4.23}$$

and we denote

$$\kappa = \frac{a_{22}}{a_{11}}, \qquad \beta = \frac{a_{12}a_{21}}{a_{11}a_{22}} \,. \tag{4.24}$$

Both eigenvalues of the matrix A have a negative real part if and only if $\det(A) > 0$. This condition can also be written as

$$\beta < 1 \,.$$

We can regard κ as a measure for the stiffness of the system, and β gives the amount of coupling between the fast and slow part of the equation. For this two-dimensional test equation we will consider asymptotic stability whereby it is required that the eigenvalues of the amplification matrix S are less than one in modulus. Similar stability considerations for 2×2 systems are found in [45] for lower order Rosenbrock methods.

The elements of the 2×2 amplification matrix S will depend on the four parameters $z_{ij} = \tau a_{ij}$, $1 \le i, j \le 2$. However, as it was shown in [45], the eigenvalues of S depend only on the determinant and trace of Z and can be written as functions of three parameters: κ, β and z_{11}. Instead of $z_{11} \le 0$ and $\beta < 1$ we will use the quantities

$$\xi = \frac{z_{11}}{1 - z_{11}}, \qquad \eta = \frac{\beta}{2 - \beta} \,, \tag{4.25}$$

which are bounded between -1 and 0, and -1 and 1, respectively.

The domains of asymptotic stability are shown in Figure 4.4. We present these domains in the (ξ, η)-plane for three values of $\kappa = 10^j$, $j = 0, 1, 2$. It is seen that the multirate RODAS will be stable if $\eta \ge 0$, whereas for $\eta < 0$ the domain of instability increases when κ gets large. The stability domains for large values of $\kappa \gg 100$ do not cover the whole region $\eta < 0$. They are similar to the domain obtained for $\kappa = 100$. Compared to the stability domains obtained for ROS2 (used with interpolation from Section 4.3) in Chapter 3, the stability domains for RODAS are smaller. However the difference is not significant. We can also see that there exist regions for which ROS2 is asymptotically unstable and RODAS is stable.

4.4.2 Kaps-Rentrop fourth-order Rosenbrock methods

We have also examined the possibility of constructing multirate methods based on the fourth-order Rosenbrock methods GRK4A and GRK4T [29]. In order

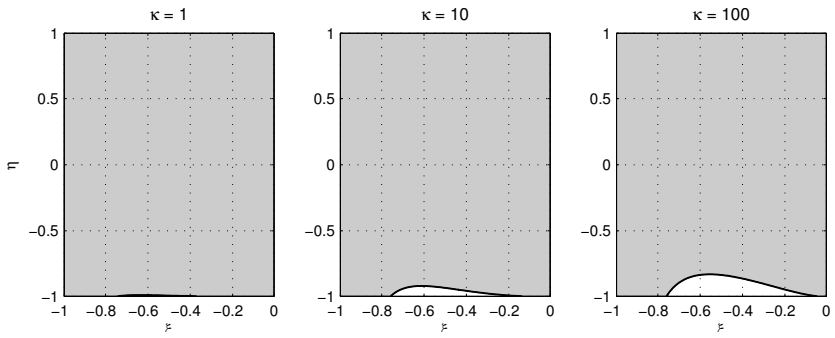

FIGURE 4.4: Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

to have a third-order interpolant, conditions (4.9) - (4.12) have to be satisfied. This set of conditions can be written as a linear system

$$Ab(\theta) = c(\theta)\,, \tag{4.26}$$

where $A \in \mathbb{R}^{4\times 4}$ is a matrix fully determined by the Rosenbrock method coefficients, $b(\theta) = [b_i(\theta)] \in \mathbb{R}^4$ is the dense output coefficients column vector and $c(\theta) = [c_i(\theta)] \in \mathbb{R}^4$ is the (4.9) - (4.12) right-hand side values column vector. For both methods GRK4A and GRK4T, the matrix A is of rank three. The second, third and fourth rows of the matrix A are linearly dependent, which also implies that the second, third and fourth elements of the column vector $c(\theta)$ have to satisfy

$$a_2 c_2(\theta) + a_3 c_3(\theta) + a_4 c_4(\theta) = 0\,, \tag{4.27}$$

with a_2, a_3 and a_4 constants dependent on the method coefficients. The relation (4.27) holds for some of the values θ (for example $\theta = 1$), however for all other values of θ it fails for both methods. Hence, we conclude that for both considered methods it is not possible to have a third-order built-in interpolant of type (4.5). The construction of such an interpolant could alternatively be achieved by adding extra stages for both methods. This would however result in an increased amount of work per step compared with the single-rate version of the original method.

4.5 Stiff source terms: the linear constant coefficient case

Use of Rosenbrock methods for problems with stiff source terms can lead to order reduction. In particular this can happen for problems with time dependent Dirichlet boundary conditions. For Rosenbrock methods, order reduction was studied for linear problems in [38]. A technique which avoids order reduction

by modifying the usual boundary values of the intermediate stages was more recently presented in [1]. During the refinement step within multirate time stepping, sub problems with time dependent boundary conditions have to be solved. Therefore, having proper local order, is of true importance for multirate schemes. In this section we aim at improving the local order of the Rosenbrock method by modifying the treatment of the source term. Using ideas from [26], we will study the order reduction for linear constant-coefficient problems.

Let us consider the linear scalar test equation

$$w'(t) = \lambda w(t) + g(t), \qquad w(0) = w_0\,, \tag{4.28}$$

where $\lambda \in \mathbb{C}$, $\mathrm{Re}\lambda \leq 0$, may be large in absolute value and also the source term may be large. However we assume that the derivatives of w are of moderate size.

The restriction to scalar problems is convenient for the notation. The results carry over to linear systems $w' = Aw + g(t)$ if A is diagonisable and well conditioned. On the other hand, the fact that only linear constant-coefficient problems are studied is a genuine restriction.

In this section, for simplicity of the expressions, it will be assumed that a time step from t_n to $t_{n+1} = t_n + \tau$ is taken. In the analysis we will derive recursions for the global errors $e_n = w(t_n) - w_n$. These recursions will be of the form

$$e_{n+1} = Se_n + d_n\,,$$

where S is the amplification factor and d_n is the local error. In case of linear test problems (4.28) we will have $S = R(z)$, where R is the stability function of the Rosenbrock method and $z = \tau\lambda$. Our aim is to derive error recursions with local errors d_n, which are independent from stiffness, so that for these recursions the derived order holds in both the non stiff and the stiff case.

4.5.1 Standard source term treatment

Error recursion

Consider an s-stage Rosenbrock method (4.2)-(4.3) with coefficients $\alpha_{ij}, \gamma_{ij}, b_j$. This leads to approximations $w_n \approx w(t_n)$ computed from

$$\begin{aligned} k_{n,i} &= z(w_n + \sum_j \beta_{ij} k_{n,j}) + \tau g(t_n + \alpha_i \tau) + \gamma_i \tau^2 g'(t_n), \quad i = 1, \ldots, s\,, \\ w_{n+1} &= w_n + \sum_j b_j k_{n,j}\,. \end{aligned} \tag{4.29}$$

Along with (4.29), we also consider the scheme with inserted exact solution values $w_n^* = w(t_n)$, $k_{n,i}^* = \tau w'(t_n + \alpha_i \tau) + \gamma_i \tau^2 w''(t_n)$. This leads to

$$\begin{aligned} k_{n,i}^* &= z(w_n^* + \sum_j \beta_{ij} k_{n,j}^* + \rho_{n,i}) + \tau g(t_n + \alpha_i \tau) + \gamma_i \tau^2 g'(t_n), \quad i = 1, \ldots, s\,, \\ w_{n+1}^* &= w_n^* + \sum_j b_j k_{n,j}^* + r_n\,, \end{aligned} \tag{4.30}$$

with residuals $\rho_{n,i}$ and r_n. For the final error recursion this choice for the exact solution values $k^*_{n,i}$ for the interior stages is not relevant. With the above choice it is the derivation of the error recursion that becomes simple.

For the analysis it is convenient to use a vector notation. Let $\boldsymbol{k}_n = [k_{n,i}] \in \mathbb{R}^s$ and denote likewise

$$\boldsymbol{G} = [\gamma_{ij}] \in \mathbb{R}^{s\times s}, \quad \boldsymbol{B} = [\beta_{ij}] \in \mathbb{R}^{s\times s},$$

$$\boldsymbol{\alpha} = [\alpha_i] \in \mathbb{R}^s, \quad \boldsymbol{\beta} = [\beta] \in \mathbb{R}^s, \quad \boldsymbol{b} = [b_i] \in \mathbb{R}^s, \quad \boldsymbol{\gamma} = [\gamma_i] \in \mathbb{R}^s, \quad \boldsymbol{e} = [1] \in \mathbb{R}^s\,.$$

Furthermore, if $\varphi : \mathbb{R} \to \mathbb{R}$, we define

$$\varphi(t_n + \boldsymbol{\alpha}\tau) = [\varphi(t_n + \alpha_i\tau)] \in \mathbb{R}^s\,.$$

This will be used for the source term g, the solution u and its derivatives.

With this notation the Rosenbrock method (4.29) can be compactly written as

$$\begin{aligned} \boldsymbol{k}_n &= z(\boldsymbol{e}w_n + \boldsymbol{B}\,\boldsymbol{k}_n) + \tau g(t_n + \boldsymbol{\alpha}\tau) + \boldsymbol{\gamma}\tau^2 g'(t_n)\,, \\ w_{n+1} &= w_n + \boldsymbol{b}^T\boldsymbol{k}_n\,. \end{aligned} \tag{4.31}$$

For the scheme with exact solution values inserted we get

$$\begin{aligned} \boldsymbol{k}^*_n &= z(\boldsymbol{e}w^*_n + \boldsymbol{B}\,\boldsymbol{k}^*_n + \boldsymbol{\rho}_n) + \tau g(t_n + \boldsymbol{\alpha}\tau) + \boldsymbol{\gamma}\tau^2 g'(t_n)\,, \\ w^*_{n+1} &= w^*_n + \boldsymbol{b}^T\boldsymbol{k}^*_n \ + r_n\,, \end{aligned} \tag{4.32}$$

with residuals $\boldsymbol{\rho}_n = [\rho_{n,i}] \in \mathbb{R}^s$ and $r_n \in \mathbb{R}$.

Expressions for these residuals are easily found by a Taylor expansion. Since we have $\boldsymbol{k}^*_n = \tau w'(t_n+\boldsymbol{\alpha}\tau)+\boldsymbol{\gamma}\tau^2 w''(t_n)$, $\lambda w(t_n+\boldsymbol{\alpha}\tau)+g(t_n+\boldsymbol{\alpha}\tau) = w'(t_n+\boldsymbol{\alpha}\tau)$ and $\lambda w'(t_n) + g'(t_n) = w''(t_n)$, it follows that

$$\begin{aligned} \boldsymbol{\rho}_n \ &= \ \frac{1}{z}\left(\tau(w'(t_n + \boldsymbol{\alpha}\tau) - g(t_n + \boldsymbol{\alpha}\tau)) + \boldsymbol{\gamma}\tau^2(w''(t_n) - g'(t_n))\right) \\ &\quad -(\boldsymbol{e}w(t_n) + \boldsymbol{B}(\tau w'(t_n + \boldsymbol{\alpha}\tau) + \boldsymbol{\gamma}\tau^2 w''(t_n))) \\ &= \ (\frac{1}{2}\boldsymbol{\alpha}^2 - \boldsymbol{B}^2\boldsymbol{e})\tau^2 w''(t_n) + \sum_{k\geq 3}\frac{1}{k!}(\boldsymbol{\alpha}^k - k\boldsymbol{B}\,\boldsymbol{\alpha}^{k-1})\tau^k w^{(k)}(t_n)\,, \end{aligned} \tag{4.33}$$

and

$$\begin{aligned} r_n \ &= \ w(t_{n+1}) - w(t_n) - \boldsymbol{b}^T(\tau w'(t_n + \boldsymbol{\alpha}\tau) + \boldsymbol{\gamma}\tau^2 w''(t_n)) \\ &= \ -\boldsymbol{b}^T\boldsymbol{\gamma}\tau^2 w''(t_n) + \sum_{k\geq 1}\frac{1}{k!}(1 - k\boldsymbol{b}^T\boldsymbol{\alpha}^{k-1})\tau^k w^{(k)}(t_n)\,, \end{aligned} \tag{4.34}$$

where $\boldsymbol{\alpha}^k = [\alpha_i^k]$ and $\boldsymbol{\alpha}^0 = \boldsymbol{e}$.

With $e_n = w^*_n - w_n$ and $\boldsymbol{\epsilon}_n = \boldsymbol{k}^*_n - \boldsymbol{k}_n$, we obtain

$$\boldsymbol{\epsilon}_n = z(\boldsymbol{e}e_n + \boldsymbol{B}\,\boldsymbol{\epsilon}_n + \boldsymbol{\rho}_n)\,,$$

$$e_{n+1} = e_n + \boldsymbol{b}^T \boldsymbol{\epsilon}_n + r_n \,.$$

Hence

$$\boldsymbol{\epsilon}_n = z(I - z\boldsymbol{B})^{-1}\boldsymbol{e}e_n + z(I - z\boldsymbol{B})^{-1}\boldsymbol{\rho}_n \,,$$

which finally gives recursion (4.5) with amplification factor $S = R(z)$,

$$R(z) = 1 + z\boldsymbol{b}^T(I - z\boldsymbol{B})^{-1}\boldsymbol{e}, \tag{4.35}$$

and local error

$$d_n = z\boldsymbol{b}^T(I - z\boldsymbol{B})^{-1}\boldsymbol{\rho}_n + r_n \,. \tag{4.36}$$

Inserting the series expansions for $\boldsymbol{\rho}_n$ and r_n, we can also write the local error as

$$\begin{aligned} d_n &= \gamma z\boldsymbol{b}^T(I - z\boldsymbol{B})^{-1}\boldsymbol{e}\tau w'(t_n) - \gamma\boldsymbol{b}^T(I - z\boldsymbol{B})^{-1}\boldsymbol{e}\tau^2 w''(t_n) \\ &+ \sum_{k\geq 1} \frac{1}{k!} H_k(z)\tau^k w^{(k)}(t_n) \end{aligned} \tag{4.37}$$

with rational functions H_k given by

$$H_k(z) = 1 - k\boldsymbol{b}^T\boldsymbol{\alpha}^{k-1} + z\boldsymbol{b}^T(I - z\boldsymbol{B})^{-1}(\boldsymbol{\alpha}^k - k\boldsymbol{B}\alpha^{k-1}) \,. \tag{4.38}$$

Stability assumptions

The stability region of the Rosenbrock method is given by the set

$$\mathcal{S} = \{z \in \mathbb{C} : |R(z)| \leq 1\} \,.$$

We assume that

$$\mathcal{S} \supset \mathbb{C}^- \,. \tag{4.39}$$

This means that the method is A-stable. In addition to this we will also assume that

$$|H_k(z)| \leq C_k \quad \text{for all } z \in \mathbb{C}^-,\, k \geq 1 \,, \tag{4.40}$$

with $C_k > 0$. Usually (4.39) implies (4.40) with $C_k > 0$ determined by the method.

Local error bounds for the stiff case

Assume that the coefficients of the Rosenbrock methods satisfy

$$\boldsymbol{b}^T\boldsymbol{\alpha}^{k-1} = \frac{1}{k} \quad \text{for } 1 \leq k \leq p_0, \quad k \neq 2 \,, \tag{4.41}$$

and

$$\boldsymbol{b}^T\boldsymbol{\beta} = \frac{1}{2} - \gamma, \quad \text{if } p_0 \geq 2, \quad k = 2 \,. \tag{4.42}$$

If the method has classical order p, then we have $p_0 \geq p$. Of course, there are many more order conditions for a method to be of order p. It will also be assumed that

$$\boldsymbol{B}\,\boldsymbol{\alpha}^{k-1} = \frac{1}{k}\boldsymbol{\alpha}^k, \qquad \text{for } 3 \leq k \leq p_1 \tag{4.43}$$

for a certain p_1 and

$$\boldsymbol{B}^2\boldsymbol{e} = \frac{1}{2}\boldsymbol{\alpha}^2, \qquad \text{if } p_1 \geq 2\,. \tag{4.44}$$

This corresponds to a so-called simplifying order condition. A method that satisfies (4.41) - (4.44) is said to have stage order $q = \min(p_0, p_1)$.

It is directly seen that these order conditions give $\mathcal{O}(\tau^{q+1})$ bounds for the residuals (4.33), (4.34) and also imply $H_k = 0$ for $k \leq q$. By the stability assumptions, it then follows that also $|d_n| = \mathcal{O}(\tau^{q+1})$. For example, for the RODAS [19], GRK4A and GRK4T [29] methods we have $q = 1$.

4.5.2 Modified source term treatment

Instead of using the source terms $g(t_n + \alpha_i\tau) + \gamma\tau g'(t_n)$ in the Rosenbrock method (4.29), we replace these by $g_{n,i}$ with $\boldsymbol{g}_n = [g_{n,i}]$ chosen as

$$\boldsymbol{g}_n = \sum_{k\geq 0} \boldsymbol{\omega}_k \tau^k g^{(k)}(t_n)\,. \tag{4.45}$$

Here $\boldsymbol{\omega}_0 = \boldsymbol{e}$ and the other $\boldsymbol{\omega}_k$ are free parameter vectors. In the vector notation, the scheme then becomes

$$\begin{aligned} \boldsymbol{k}_n &= \tau(\lambda \boldsymbol{e} w_n + \lambda \boldsymbol{B}\,\boldsymbol{k}_n + \sum_{k\geq 0} \boldsymbol{\omega}_k \tau^k g^{(k)}(t_n))\,, \\ w_{n+1} &= w_n + \boldsymbol{b}^T \boldsymbol{k}_n\,. \end{aligned} \tag{4.46}$$

As before, we also consider a perturbed scheme with exact solution values inserted,

$$\begin{aligned} \boldsymbol{k}^*_n &= \tau(\lambda \boldsymbol{e} w^*_n + \lambda \boldsymbol{B}\,\boldsymbol{k}^*_n + \sum_{k\geq 0} \boldsymbol{\omega}_k \tau^k g^{(k)}(t_n) + \lambda\boldsymbol{\rho}_n)\,, \\ w^*_{n+1} &= w^*_n + \boldsymbol{b}^T \boldsymbol{k}^*_n + r_n\,. \end{aligned} \tag{4.47}$$

We take again $w^*_n = w(t_n)$. For $\boldsymbol{k}^*_n$ it is now convenient to choose

$$\boldsymbol{k}^*_n = \sum_{k\geq 0} \boldsymbol{\omega}_k \tau^{k+1} w^{(k+1)}(t_n)\,.$$

This gives residuals

$$\boldsymbol{\rho}_n = \sum_{k\geq 1} (\boldsymbol{\omega}_k - \boldsymbol{B}\,\boldsymbol{\omega}_{k-1}) \tau^k w^{(k)}(t_n)\,, \tag{4.48}$$

$$r_n = \sum_{k\geq 1} (\frac{1}{k!} - \boldsymbol{b}^T \boldsymbol{\omega}_{k-1}) \tau^k w^{(k)}(t_n)\,. \tag{4.49}$$

The requirement $\boldsymbol{\rho}_n, r_n = \mathcal{O}(\tau^{q+1})$ thus leads to the conditions

$$\boldsymbol{\omega}_k = \boldsymbol{B}\boldsymbol{\omega}_{k-1}, \quad \boldsymbol{b}^T \omega_{k-1} = \frac{1}{k!} \quad (k = 1, \ldots, q)\,, \tag{4.50}$$

that is,

$$\boldsymbol{\omega}_k = \boldsymbol{B}^k \boldsymbol{e}, \quad \boldsymbol{b}^T \boldsymbol{B}^{k-1} \boldsymbol{e} = \frac{1}{k!} \quad (k = 1, \ldots, q)\,. \tag{4.51}$$

Note that if a method is of order p for non-stiff problems, then the condition

$$\boldsymbol{b}^T \boldsymbol{B}^{k-1} \boldsymbol{e} = \frac{1}{k!}$$

holds for all $k = 1, \ldots, p$. Therefore, in order to have a method of order p for stiff problems, both conditions (4.51) should be fulfilled and we still have to require

$$\boldsymbol{\omega}_k = \boldsymbol{B}^k \boldsymbol{e}, \quad (k = 1, \ldots, p)\,. \tag{4.52}$$

The source term $g(t_n + \boldsymbol{c}\tau)$ can also be replaced by a more general series expansion

$$\boldsymbol{g}_n = \sum_{k \geq 0} \boldsymbol{Q}_k \tau^k g^{(k)}(t_n + \boldsymbol{\mu}_k \tau)\,, \tag{4.53}$$

where $\boldsymbol{Q}_k$ and $\boldsymbol{\mu}_k$ are free parameter matrices and vectors respectively. In this case the condition (4.52) becomes

$$\sum_{l=0}^{k} \frac{1}{(k-l)!} \boldsymbol{Q}_l \boldsymbol{\mu}_l^{k-l} = \boldsymbol{B}^k \boldsymbol{e} \quad (k = 1, \ldots, q)\,. \tag{4.54}$$

While (4.52) requires the first p derivatives $g^{(k)}(t_n), k = 1, \ldots, p$, use of the source term in the more general form (4.53) may allow less derivatives.

Example 4.5.1 In order to recover one order for stiff problems, that is, to increase the stage order by one unit, one can use the source term modification of type (4.45)

$$\boldsymbol{g}_n = \sum_{k=0}^{2} \boldsymbol{B}^k \boldsymbol{e} \tau^k g^{(k)}(t_n)\,,$$

which uses the first two derivatives of the source function $g(t)$. One can also use the modification of type (4.53)

$$\boldsymbol{g}_n = \boldsymbol{e} g(t_n) + \boldsymbol{B}\tau g'(t_n + \boldsymbol{\beta}\tau) \tag{4.55}$$

which only requires the value of the first derivative $g'(t)$.

To recover two orders, again, one can choose between

$$\boldsymbol{g}_n = \sum_{k=0}^{3} \boldsymbol{B}^k \boldsymbol{e} \tau^k g^{(k)}(t_n) \tag{4.56}$$

and

$$\boldsymbol{g}_n = \boldsymbol{e} g(t_n) + \boldsymbol{B}\boldsymbol{e}\tau g'(t_n) + \boldsymbol{B}^2\tau^2 g''(t_n + \boldsymbol{\beta}\tau)\,. \tag{4.57}$$

Formula (4.56) cannot be modified such that only the functions $g(t)$ and $g'(t)$ are used. The attempt to replace (4.56) with

$$\boldsymbol{g}_n = g(t_n + \boldsymbol{\xi}_1\tau) + \boldsymbol{P}\,\boldsymbol{e}\tau g'(t_n + \boldsymbol{\xi}_2\tau)$$

leads to an unsolvable system. □

4.5.3 Effect on the convergence for non-stiff problems

For non-stiff problems (4.28), where λ is of moderate size, and using our modified source term (4.45), we obtain the following expansion for the local error

$$\begin{aligned} d_n \;=\; & \sum_{k\geq 1}\left(\frac{1}{k!} - \boldsymbol{b}^T\boldsymbol{\omega}_{k-1}\right)\tau^k w^{(k)}(t_n) \\ & + \sum_{k\geq 2}\sum_{j=1}^{k-1}\lambda^{k-j}\boldsymbol{b}^T\boldsymbol{B}^{k-j-1}(\boldsymbol{\omega}_j - \boldsymbol{B}\boldsymbol{\omega}_{j-1})\tau^k w^{(j)}(t_n)\,. \end{aligned} \tag{4.58}$$

We require that this remains $\mathcal{O}(\tau^{p+1})$, that is, we want the modification (4.45) of the source term to be such that the classical order of consistency p is recovered. We are thus left with the order conditions

$$\boldsymbol{b}^T\boldsymbol{\omega}_{k-1} = \frac{1}{k!}, \qquad \boldsymbol{b}^T\boldsymbol{B}^{k-j-1}(\boldsymbol{\omega}_j - \boldsymbol{B}\boldsymbol{\omega}_{j-1}) = 0, \qquad (1 \leq j < k \leq p)\,. \tag{4.59}$$

Since $\boldsymbol{\omega}_0 = \boldsymbol{e}$ and $\boldsymbol{b}^T B^{k-1}\boldsymbol{e} = \frac{1}{k!}$ for $l \leq p$, it follows that these order conditions are covered by

$$\boldsymbol{b}^T\boldsymbol{B}^{k-j-1}\boldsymbol{\omega}_j = \frac{1}{k!} \qquad (1 \leq j < k \leq p)\,. \tag{4.60}$$

The standard form of the source term can be expanded as

$$g(t_n + \boldsymbol{\alpha}\tau) + \boldsymbol{\gamma}\tau g'(t_n) = \boldsymbol{e} g(t_n) + \boldsymbol{\beta}\tau g'(t_n) + \sum_{k\geq 2}\frac{1}{k!}\boldsymbol{\alpha}^k\tau^k g^{(k)}(t_n)\,, \tag{4.61}$$

which gives

$$\boldsymbol{\omega}_0 = \boldsymbol{e}, \qquad \boldsymbol{\omega}_1 = \boldsymbol{\beta}, \qquad \boldsymbol{\omega}_k = \frac{1}{k!}\boldsymbol{\alpha}^k, \qquad k \geq 2\,. \tag{4.62}$$

We know that the use of the source term in the standard form leads to consistency of order p. Thus the coefficients (4.62) satisfy condition (4.60).

If we consider

$$\boldsymbol{\omega}_k = \boldsymbol{B}^k\boldsymbol{e}, \qquad (k = 1, \ldots, p) \tag{4.63}$$

then

$$\boldsymbol{b}^T\boldsymbol{B}^{k-j-1}\boldsymbol{\omega}_j = \boldsymbol{b}^T\boldsymbol{B}^{k-j-1}\boldsymbol{B}^j\boldsymbol{e} = \boldsymbol{b}^T\boldsymbol{B}^{k-1}\boldsymbol{e} = \frac{1}{k!}\,. \tag{4.64}$$

This shows that the choice (4.63) helps us to recover the order of consistency p for stiff problems and that it also does not affect the order of consistency for non-stiff problems. If, however, (4.60) holds just for k with $1 \leq j < k < p$, then the order of consistency for non-stiff problems can be lost. For example, for fourth-order Rosenbrock methods, we loose one order if we use (4.55) for non-stiff problems and we preserve the order in case of (4.57).

4.6 Numerical experiments

In this section we present numerical results for four test problems. In the first test problem we consider the order behavior of the RODAS method. Results for the standard and the modified source term treatment are presented. Along with the single-rate time integration with time steps of size τ we perform the dual-rate time integration, where after each time step of size 2τ the solution is refined at a fixed spatial region by taking two smaller time steps of size τ. For the other three test problems we use the self-adjusting multirate time stepping strategy presented in Chapter 1. Given a global time step τ, we compute a first, tentative approximation at the new time level for all components. For those components for which the error estimator indicates that smaller steps are needed, the computation is redone with $\frac{1}{2}\tau$. The refinement is continued recursively with local time steps $2^{-l}\tau$, until the error estimator is below a prescribed tolerance for all components. The numerical results obtained for the RODAS method are compared with those obtained using second-order ROS2 method [47]. For these tests we also use the source term treatment modifications suggested in Section 4.5. These modifications used for ROS2 give similar results with those obtained using the standard source term treatment for these problems.

4.6.1 A linear parabolic example

As a test model we consider the parabolic equation (also used in [25])

$$u_t + au_x \;=\; du_{xx} - cu + g(x,t)\,, \tag{4.65a}$$

for $0 < t < T = 0.4$, $-1 < x < 1$, with initial- and boundary conditions

$$u(x,0) = 0\,, \qquad u(-1,t) = 0\,, \qquad u(1,t) = 0\,. \tag{4.65b}$$

The constants and source term are taken as

$$a = 10\,, \quad d = 1\,, \quad c = 10^2\,, \quad g(x,t) = 10^3 \cos(\frac{1}{2}\pi x)^{100} \sin(\pi t)\,. \tag{4.65c}$$

The solution at the end time $t = 0.4$ is illustrated in Figure 2.4 in Chapter 2.

Semi-discretization with second-order differences on a uniform spatial grid with m points and mesh width $h = 2/(m+1)$, leads to an ODE system of the form (4.1). We use for this test $m = 400$, and the temporal refinements are

taken for the components corresponding to spatial grid points $x_j \in [-0.2, 0.2]$. (Spatial grid refinements are not considered here; we use the semi-discrete system just as an ODE example.) We solve the problem with the RODAS method described in Section 4.4.1.

Tables 4.1 and 4.2 show the maximum errors at $t = T$ with respect to a time-accurate ODE solution. The results are given for the single-rate case with uniform time steps $\tau = T/N$ and for the multirate case, where each time step 2τ is followed by two locally refined steps τ on part of the spatial domain. For both cases the standard and the modified source term treatment described in Section 4.5 are considered.

TABLE 4.1: Errors and orders for problem (4.65), single-rate case

	Single-rate without correction		Single-rate with correction	
N	error	order	error	order
10	$3.08 \cdot 10^{-5}$		$3.01 \cdot 10^{-5}$	
20	$3.48 \cdot 10^{-6}$	3.14	$1.35 \cdot 10^{-6}$	4.47
40	$3.60 \cdot 10^{-7}$	3.27	$6.06 \cdot 10^{-8}$	4.48
80	$3.45 \cdot 10^{-8}$	3.38	$2.92 \cdot 10^{-9}$	4.37
160	$3.07 \cdot 10^{-9}$	3.49	$1.55 \cdot 10^{-10}$	4.23

TABLE 4.2: Errors and orders for problem (4.65), multirate case

	Multirate without correction		Multirate with correction	
N	error	order	error	order
10	$7.95 \cdot 10^{-4}$		$8.86 \cdot 10^{-4}$	
20	$3.05 \cdot 10^{-5}$	4.70	$3.17 \cdot 10^{-5}$	4.80
40	$1.96 \cdot 10^{-6}$	3.95	$8.25 \cdot 10^{-7}$	5.26
80	$3.46 \cdot 10^{-7}$	2.50	$2.36 \cdot 10^{-8}$	5.12
160	$7.14 \cdot 10^{-8}$	2.27	$1.13 \cdot 10^{-9}$	4.38

The refinement region $-0.2 \leq x_j \leq 0.2$ was only chosen for test purposes; it is clear from Figure 2.4 that it is not a very good choice. Tables 4.1 and 4.2 show that for this example we get order reduction for both single-rate and multirate cases when we use the standard formulation of the Rosenbrock method. With the modification from Section 4.5 we recover the fourth order of the RODAS method. One can also see that the errors for the multirate case are somewhat

larger than the corresponding errors for the single-rate case. This can be explained by the fact that the solution is active outside the refinement interval and integration with one time step of size 2τ is less accurate than the integration with two time steps of size τ for this spatial region.

4.6.2 The inverter chain problem

As a second test example we consider the inverter chain problem from [3]. The model for m inverters consists of the equations

$$\begin{cases} w_1'(t) = U_{\text{op}} - w_1(t) - \Upsilon g\Big(u_{\text{in}}(t), w_1(t)\Big)\,, \\ w_j'(t) = U_{\text{op}} - w_j(t) - \Upsilon g\Big(w_{j-1}(t), w_j(t)\Big)\,, \qquad j = 2, \ldots, m\,, \end{cases} \tag{4.66a}$$

where

$$g(u,v) = \Big(\max(u - U_{\text{thres}}, 0)\Big)^2 - \Big(\max(u - v - U_{\text{thres}}, 0)\Big)^2. \tag{4.66b}$$

The coefficient Υ serves as stiffness parameter. We solve the problem for a chain of $m = 500$ inverters with $\Upsilon = 100$, $U_{\text{thres}} = 1$ and $U_{\text{op}} = 5$, over the time interval $[0, T]$, $T = 130$. The initial condition is

$$w_j(0) = 6.247 \cdot 10^{-3} \ \text{ for } j \text{ even}, \quad w_j(0) = 5 \ \text{ for } j \text{ odd}. \tag{4.66c}$$

The input signal is given by

$$u_{\text{in}}(t) = \begin{cases} t - 5 & \text{for} \quad 5 \le t \le 10\,, \\ 5 & \text{for} \quad 10 \le t \le 15\,, \\ \frac{5}{2}(17 - t) & \text{for} \quad 15 \le t \le 17\,, \\ 0 & \text{otherwise.} \end{cases} \tag{4.66d}$$

An illustration for some even components of the solution is given in Figure 1.8 in Chapter 1.

In Table 4.3 the maximal errors over all components and all times t_n (measured with respect to an accurate reference solution) are presented for several tolerances with the single-rate scheme (without local temporal refinements) and the multirate strategy. As a measure for the amount of work we consider the total number of linear systems that had to be solved. In addition, the CPU times (in seconds) are given. In Figure 4.5 the CPU-error diagram is presented, where the values for the ROS2 method are taken from [47]. It shows that the multirate RODAS method is more efficient than the multirate version of ROS2.

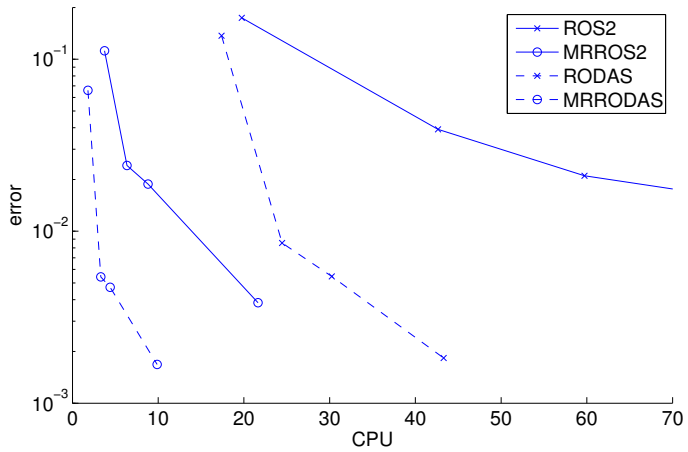

FIGURE 4.5: CPU-error diagram for problem (4.66).

TABLE 4.3: Absolute maximal errors, work amount and CPU time with different tolerances for the inverter chain problem, RODAS

	Single-rate			Multirate		
tol	error	work	CPU	error	work	CPU
$5 \cdot 10^{-4}$	$1.37 \cdot 10^{-1}$	49554000	17.39	$6.60 \cdot 10^{-2}$	2686848	1.81
$1 \cdot 10^{-4}$	$8.55 \cdot 10^{-3}$	69705000	24.46	$5.43 \cdot 10^{-3}$	5120184	3.31
$5 \cdot 10^{-5}$	$5.46 \cdot 10^{-3}$	85935000	30.25	$4.72 \cdot 10^{-3}$	6742536	4.40
$1 \cdot 10^{-5}$	$1.83 \cdot 10^{-3}$	125031000	43.92	$1.68 \cdot 10^{-3}$	12570852	9.88

4.6.3 An ODE system obtained from semi-discretization: a reaction-diffusion problem with traveling wave solution

For our third test problem we consider the semi-discrete system obtained from the reaction-diffusion equation

$$u_t \;=\; \epsilon u_{xx} + \gamma u^2(1-u), \tag{4.67}$$

for $0 < x < L$, $0 < t \leq T$. The initial- and boundary conditions are given by

$$u_x(0,t) \;=\; u_x(L,t) \;=\; 0\,, \qquad u(x,0) \;=\; \left(1+\mathrm{e}^{\lambda(x-1)}\right)^{-1}, \tag{4.68}$$

where $\lambda = \frac{1}{2}\sqrt{2\gamma/\epsilon}$. If the spatial domain had been the whole real line, then the initial profile would have given the traveling wave solution $u(x,t) = u(x - \alpha t, 0)$ with velocity $\alpha = \frac{1}{2}\sqrt{2\gamma\epsilon}$. In our problem, with homogeneous Neumann boundary conditions, the solution will still be very close to this traveling wave, provided the end time is sufficiently small so that the wave front does not come close to the boundaries. The parameters are taken as $\gamma = 1/\epsilon = 100$ and $L = 5$, $T = 3$. In space we used a uniform grid of $m = 1000$ points and standard second-order differences, leading to an ODE system in $\mathbb{R}^{1000}$. An illustration of the semi-discrete solution at various times is given in Figure 1.4 with (spatial) components horizontally.

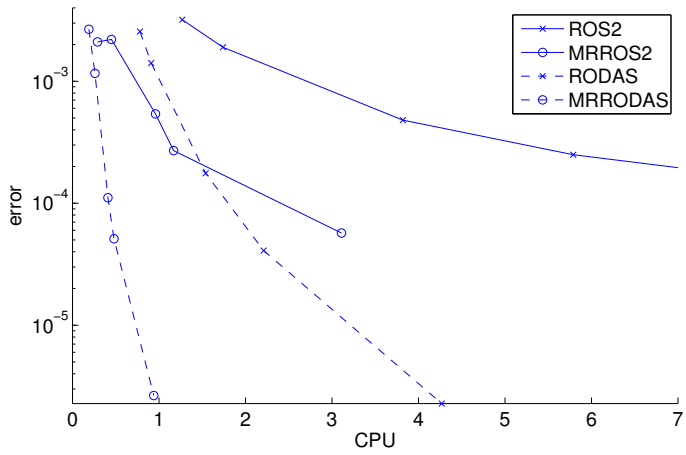

FIGURE 4.6: CPU-error diagram for problem (4.67).

In Table 4.4 the errors (in the maximum norm with respect to the reference ODE solution at time T), the amount of work (number of linear systems that had to be solved) and CPU time (in seconds) are presented for different tolerances. From these results it is seen that a substantial improvement in amount of work is obtained for this problem. For the single-rate scheme, the amount of work is almost six times larger. In terms of CPU time we get a speed-up factor four approximately. Moreover, the error behavior of the multirate scheme is very good. We have roughly a proportionality of the errors and tolerances, and the errors of the multirate scheme are approximately the same as for the single-rate scheme.

In Figure 4.6 the CPU-error diagram is presented, where the values for the ROS2 method are taken from [47]. It shows that the multirate RODAS method is more efficient than the multirate version of ROS2.

TABLE 4.4: Absolute maximal errors, work amount and CPU time with different tolerances for the traveling wave problem, RODAS

	Single-rate			Multirate		
tol	error	work	CPU	error	work	CPU
$1 \cdot 10^{-3}$	$2.56 \cdot 10^{-3}$	1213212	0.78	$2.67 \cdot 10^{-3}$	317648	0.19
$5 \cdot 10^{-4}$	$1.41 \cdot 10^{-3}$	1417416	0.91	$1.16 \cdot 10^{-3}$	330156	0.26
$1 \cdot 10^{-4}$	$1.76 \cdot 10^{-4}$	2396394	1.54	$1.11 \cdot 10^{-4}$	482694	0.41
$5 \cdot 10^{-5}$	$4.09 \cdot 10^{-5}$	3417414	2.21	$5.11 \cdot 10^{-5}$	571782	0.48
$1 \cdot 10^{-5}$	$2.28 \cdot 10^{-6}$	6582576	4.27	$2.65 \cdot 10^{-6}$	1030740	0.94

4.6.4 Transmission line problem

The M-dimensional transmission line circuit (obtained from A. Verhoeven, private communication) can be described by the system

$$\begin{cases} v_k'(t) &= \frac{1}{c}\left(i_{k+1}(t) - i_k(t)\right) , \\ i_k'(t) &= \frac{1}{l}\left(v_k(t) - v_{k-1}(t) - r i_k(t)\right) , \end{cases} \tag{4.69a}$$

for $k = 1, \ldots, M$, where $i_{M+1}(t) = 0$, $v_0(t) = v_{in}(t) + 10^3 i_1(t)$,

$$v_{in}(t) = \begin{cases} 1 & \text{if } t > 10^{-11} \\ 10^{11} t & \text{if } t \le 10^{-11} \end{cases}$$

and

$$v_k(0) = 0, \qquad i_k(0) = 0\,, \quad k = 1, \ldots, M\,. \tag{4.69b}$$

We solve the problem for $M = 100$ with $r = 0.35$, $c = 4 \times 10^{-13}$ and $l = 10^{-9}$. An illustration of the solution for some of the components is given in Figure 4.7.

For the numerical test, the multirate method based on the second-order ROS2 described in Chapter 1 and the multirate method based on the fourth-order RODAS are used. In Tables 4.5 and 4.6, the errors at output time $T = 10^{-9}$, measured in the maximum norm over time and components with respect to an accurate reference solution, together with the amount of work (number of linear systems to be solved) and CPU time (in seconds), are presented for different values of tolerance for the single-rate and the multirate strategies. For this test we do not get much improvement when using the multirate strategy. For the single-rate scheme, the amount of work is almost two times larger.

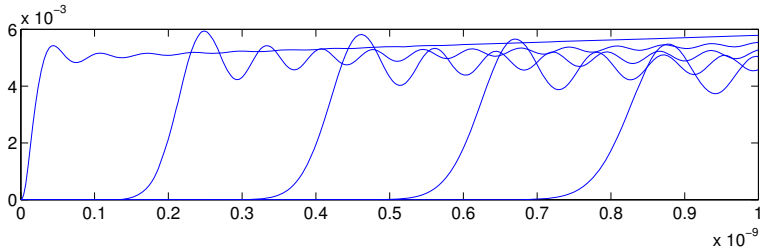

FIGURE 4.7: Solution components v_k, $k = 1, 10, 20, 30, 40$, for problem (4.69).

TABLE 4.5: Errors, work amount and CPU time for problem (4.69), ROS2

	Single-rate			Multirate		
tol	error	work	CPU	error	work	CPU
$1 \cdot 10^{-4}$	$5.49 \cdot 10^{-4}$	38800	0.05	$4.27 \cdot 10^{-4}$	20984	0.04
$5 \cdot 10^{-5}$	$3.08 \cdot 10^{-4}$	55600	0.07	$2.66 \cdot 10^{-4}$	28816	0.05
$1 \cdot 10^{-5}$	$6.88 \cdot 10^{-5}$	122400	0.14	$6.62 \cdot 10^{-5}$	66669	0.09
$5 \cdot 10^{-6}$	$3.42 \cdot 10^{-5}$	174000	0.23	$3.67 \cdot 10^{-5}$	96052	0.16
$1 \cdot 10^{-6}$	$6.92 \cdot 10^{-6}$	384800	0.44	$5.60 \cdot 10^{-6}$	206648	0.31

TABLE 4.6: Errors, work amount and CPU time for problem (4.69), RODAS

	Single-rate			Multirate		
tol	error	work	CPU	error	work	CPU
$1 \cdot 10^{-4}$	$1.24 \cdot 10^{-4}$	66000	0.07	$1.32 \cdot 10^{-4}$	38832	0.06
$5 \cdot 10^{-5}$	$5.26 \cdot 10^{-5}$	82800	0.09	$3.94 \cdot 10^{-5}$	49608	0.07
$1 \cdot 10^{-5}$	$5.30 \cdot 10^{-6}$	139200	0.15	$5.40 \cdot 10^{-6}$	84684	0.12
$5 \cdot 10^{-6}$	$2.12 \cdot 10^{-6}$	174000	0.23	$3.06 \cdot 10^{-6}$	103409	0.16
$1 \cdot 10^{-6}$	$4.47 \cdot 10^{-7}$	288000	0.32	$5.45 \cdot 10^{-7}$	164544	0.25

Improvement in CPU time is smaller due to the extra work required for the automatic partitioning.

In general, the execution time of a program based on our multirate strategy is not greater than that of a program based on the single-rate strategy. In

the case of multirating not leading to an improvement in work, the multirate strategy automatically takes the same time steps as in the single-rate strategy.

In Figure 4.8 the CPU-error diagram is presented. It shows that the multirate RODAS method is more efficient than the multirate version of ROS2.

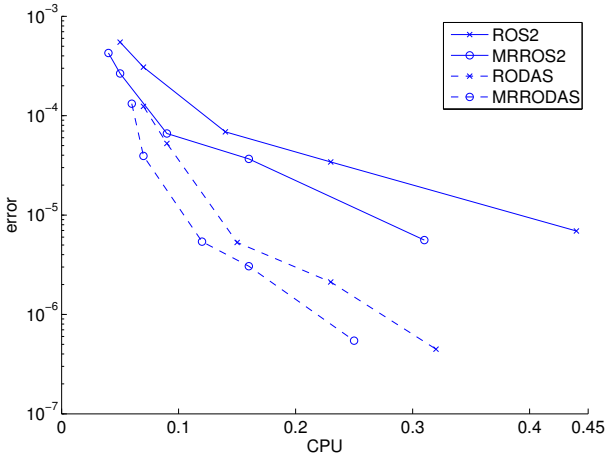

FIGURE 4.8: CPU-error diagram for problem (4.69).

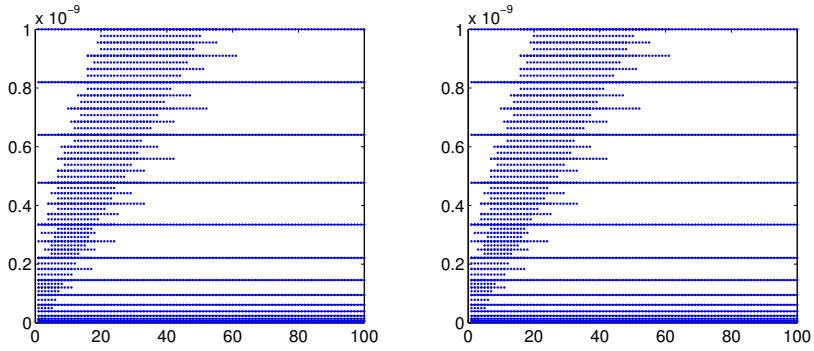

FIGURE 4.9: Component-time grid (v_k left and i_k right) for problem (4.69).

In Figure 4.9 the component-time grids are shown on which the solution was calculated using the multirate RODAS method with tolerance value $tol = 2 \cdot 10^{-3}$.

In principle these two grids can be different. However, in the experiments they are practically the same.

4.7 Conclusions

In this chapter we discussed the main aspects of the construction of higher-order multirate methods.

As seen from the numerical tests, improper treatment of stiff source terms and use of lower-order interpolants can lead to an order reduction where we obtain a lower order of consistency than for non-stiff problems.

We presented a strategy of avoiding the order reduction for problems with a stiff source term. This strategy helps us to recover the order of consistency for stiff problems and does not affect the order of consistency for non-stiff problems.

A multirate method based on the fourth-order Rosenbrock method RODAS and its third-order dense output was designed. The multirate RODAS method showed good results in the numerical experiments and is clearly more efficient than the considered second-order multirate methods.

Acknowledgments

We would like to thank A. Verhoeven and E. J. W. ter Maten from Eindhoven Univ. of Technology and NXP Semiconductors for suggesting the transmission line and inverter chain test problems.

4.8 Appendix

In Table 4.7 we present the coefficients of the RODAS method, which were derived following [19, pp. 421]. The coefficients of the built-in dense output of the RODAS are presented in Table 4.8. These coefficients were chosen to satisfy the third-order conditions (4.9)-(4.12), the first fourth-order condition (4.13) and the condition $b_6(\theta) = \gamma\theta$, see [19].

TABLE 4.7: Coefficients of the RODAS method

$\alpha_{21} = 0.386$	$\alpha_{31} = 0.146074707525418$	$\alpha_{32} = 0.063925292474582$
$\alpha_{41} = -0.330811503667722$	$\alpha_{42} = 0.711151025168282$	$\alpha_{43} = 0.24966047849944$
$\alpha_{51} = -4.552557186318003$	$\alpha_{52} = 1.710181363241322$	$\alpha_{53} = 4.014347332103150$
$\alpha_{54} = -0.171971509026469$	$\alpha_{61} = 2.428633765466978$	$\alpha_{62} = -0.382748733764781$
$\alpha_{63} = -1.855720330929574$	$\alpha_{64} = 0.559835299227375$	$\alpha_{65} = 0.25$
$\gamma = 0.25$		
$\gamma_{21} = -0.3543$	$\gamma_{31} = -0.133602505268175$	$\gamma_{32} = -0.012897494731825$
$\gamma_{41} = 1.526849173006459$	$\gamma_{42} = -0.533656288750454$	$\gamma_{43} = -1.279392884256$
$\gamma_{51} = 6.981190951784981$	$\gamma_{52} = -2.092930097006103$	$\gamma_{53} = -5.870067663032724$
$\gamma_{54} = 0.731806808253845$	$\gamma_{61} = -2.080189494180926$	$\gamma_{62} = 0.59576235567668$
$\gamma_{63} = 1.701617798267255$	$\gamma_{64} = -0.088514519835879$	$\gamma_{65} = -0.378676139927128$
$b_1 = 0.348444271286054$	$b_2 = 0.213013621911897$	$b_3 = -0.154102532662319$
$b_4 = 0.471320779391497$	$b_5 = -0.128676139927129$	$b_6 = 0.25$

TABLE 4.8: Coefficients of the RODAS dense output

$b_{10} = 1.158234160966162$	$b_{11} = 3.888756124907816$	$b_{12} = -9.858437647569822$
$b_{13} = 5.159891632981919$	$b_{20} = 2.048767778074541$	$b_{21} = -4.936277941843626$
$b_{22} = 4.578307037111220$	$b_{23} = -1.477783251430241$	$b_{30} = -1.392687054381870$
$b_{31} = -1.897781380424416$	$b_{32} = 7.357213793345069$	$b_{33} = -4.220847891201125$
$b_{40} = -0.945903133634689$	$b_{41} = 3.525328088642974$	$b_{42} = -2.327663658815888$
$b_{43} = 0.219559483199102$	$b_{50} = -0.118411751024145$	$b_{51} = -0.580024891282749$
$b_{52} = 0.250580475929419$	$b_{53} = 0.319180026450346$	$b_{60} = 0.25$
$b_{61} = 0$	$b_{62} = 0$	$b_{63} = 0$