Multirate Numerical Integration for Ordinary Differential Equations

Valeriu Savcenco

Printed and bound by Ponsen & Looijen bv.

CIP-DATA LIBRARY UNIVERSITEIT VAN AMSTERDAM

Valeriu Savcenco

Multirate Numerical Integration for Ordinary Differential Equations/ by Valeriu Savcenco.– Amsterdam: Universiteit van Amsterdam, 2007.– Proefschrift.

ISBN 90-6196-543-8

Subject headings: Multirate time stepping / Local time stepping / Ordinary differential equations / Stiff differential equations / Asymptotic stability / High-order Rosenbrock methods / Partitioned Runge-Kutta methods / Monotonicity / TVD / Stability / Convergence.

Multirate Numerical Integration for Ordinary Differential Equations

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. D.C. van den Boom
ten overstaan van een door
het college voor promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op dinsdag 15 januari 2008, te 10:00 uur

door

Valeriu Savcenco

geboren te Chişinău, Moldavië

Promotiecommissie

Promotor:	prof.dr. J.G. Verwer
Co-promotor:	dr. W. Hundsdorfer
Overige leden:	prof.dr. R.P. Stevenson
	dr. J. Brandts
	prof.dr. P.W. Hemker
	prof.dr. R.M.M. Mattheij
	dr. E.J.W. ter Maten
	prof.dr. C. Vuik

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

NWO
Nederlandse Organisatie voor Wetenschappelijk Onderzoek

Thomas Stieltjes Institute
for Mathematics

CWI

Het onderzoek dat tot dit proefschrift heeft geleid werd mede mogelijk gemaakt door een Peter Paul Peterichbeurs –verstrekt door de Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO)– en het Centrum voor Wiskunde en Informatica (CWI).

To the memory of my mother
To my father and my sister
To the future

Contents

Preface

Many disciplines, such as physics, the natural and biological sciences, engineering, economics and the financial sciences frequently give rise to problems that need mathematical modelling for their solutions. Examples vary in scale from the behavior of cells in biology, to the behavior of electrical circuits, to flows and combustion processes in a jet engine, to the formation and development of galaxies. Standard practice is that the solution of the mathematical models are not known in closed, analytic form, and hence must be computed approximately by means of algorithms and software from numerical mathematics and scientific computing. Numerical mathematics and scientific computing therefore are of great relevance in modern applied sciences and are the crucial tools for their qualitative and quantitative analysis.

This thesis records the numerical mathematics research I conducted between February 2004 and February 2008 in the Modeling, Analysis and Simulation (MAS) department of the Centrum voor Wiskunde en Informatica (CWI) in Amsterdam. It deals with the development of multirate time stepping techniques for systems of ordinary differential equations. Multirate methods allow one to use large time steps for slowly varying components, and small steps for rapidly varying ones. Numerical experiments confirm that the efficiency of time integration methods can be significantly improved by using multirate methods.

The thesis consists of five chapters preceded by an introduction and followed by a summary. The chapters are based on published and submitted papers. Details are listed below:

1. Chapter 1 is based on the paper by V. Savcenco, W. Hundsdorfer and J.G. Verwer, entitled *A multirate time stepping strategy for stiff ordinary differential equations*, published in BIT 47, pages 137-155, 2007.

2. Chapter 2 is based on the paper by W. Hundsdorfer and V. Savcenco, entitled *Analysis of a multirate theta-method for stiff ordinary differential equations*, accepted for publication in Applied Numerical Mathematics.

3. Chapter 3 is based on the paper by V. Savcenco, entitled *Comparison of the asymptotic stability properties for two multirate strategies*, accepted for publication in Journal of Computational and Applied Mathematics.

4. Chapter 4 is based on the paper, entitled *Construction of high-order multirate Rosenbrock methods for stiff ODEs*, by V. Savcenco, to be submitted.

5. Chapter 5 is entitled *Analysis of explicit multirate and partitioned Runge-Kutta schemes for conservation laws*, a joint work with W. Hundsdorfer

and A. Mozartova, to be submitted.

The introductory chapter is meant to help unspecialized readers to understand the motivation and the subject, as well as to outline the content of the whole thesis. The summary will summarize the conclusions that we have pointed out in the thesis.

Valeriu Savcenco

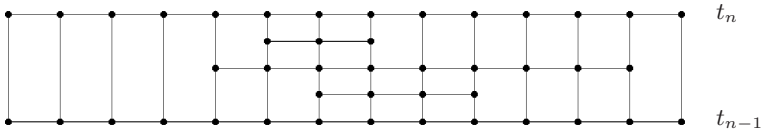
Amsterdam, October 2007

Introduction

Numerous phenomena from different areas of science and technology are modelled by systems of ordinary differential equations (ODEs). ODEs describe the motion of a body by its position and velocity; the evolution of the current in an electrical circuit; the change of the temperature of an object in a given environment; and even the dynamics of the price of a stock. In addition, some methods in numerical partial differential equations (PDEs) convert the partial differential equation into an ordinary differential equation system, which then must be solved. Most ODEs cannot be solved analytically, in which case an approximation to the solution is found by applying numerical integration methods.

For the numerical solution of systems of ODEs there are many methods available; see for example the text books of Butcher [7], Hairer et al. [18, 19], Lambert [32]. These methods use time steps that are varying in time, but are constant over the components. However, there are many problems of practical interest, where the temporal variations have different time scales for different sets of the components. For example, cellular phones consist of coupled digital and analogue sub-circuits, which operate in nano- and micro-seconds, respectively. The motion of the particles around a star, which attracts mass from a secondary star, in astrophysics is described by a large system of ordinary differential equations. In this system the components, that correspond to the particles near the center, are much faster than those corresponding to the distant ones. To exploit these local time scale variations, one needs *multirate methods* that use different, local time steps over the components. In these methods big time steps are used for the slow components and small time steps are used for the fast ones.

Also the components can have more levels of activity. For example, there can be slow, intermediate and fast components.



In the above figure we present a time slab (with components horizontally and time vertically) of size $\Delta t_n = t_n - t_{n-1}$, in which an approximation to the solution at time t_n is computed. In this example, time steps of size $\frac{1}{4}\Delta t_n$, $\frac{1}{2}\Delta t_n$ and Δt_n are used depending on the activity of the components. The activity of the components can also change in time, the slow components can become

active and the active components can become slow.

The major aim of this thesis is to design, analyze and test multirate methods for the numerical solution of ODEs. Different local time steps require care since coupled components then may need information at different time levels. These values can be obtained by interpolation, extrapolation or dense output. The choice of interpolant is crucial, because it has direct influence on the stability and the order of convergence of the multirate method.
The thesis consists of five chapters, preceded by this introduction, and ends with a summary.

Chapter 1. In the first chapter, we introduce a self-adjusting multirate time stepping strategy for the numerical solution of ODEs. The step size for a particular system component is determined by the local temporal variation of the solution, in contrast to the use of a single step size for the whole set of components as in the traditional (single-rate) methods. For a given global time step $\Delta t_n = t_n - t_{n-1}$, a tentative approximation at the time level t_n for all components is computed first. The components, for which an error estimator indicates that smaller steps are needed, are computed again with halved step size $\frac{1}{2}\Delta t_n$. The refinement is recursively continued until an error estimator is below a prescribed tolerance for all components. The size of the time slabs Δt_n is determined automatically, while advancing in time, in a way which gives minimal amount of work per time unit without loosing accuracy. The performance of the strategy is demonstrated on the basis of three numerical examples. A second-order Rosenbrock method with an embedded first-order method is used as basic time stepping method. Numerical experiments confirm that the efficiency of time integration can be significantly improved by the use of multirate methods.

Chapter 2. The second chapter contains a study of a simple multirate scheme, consisting of the θ-method with one level of temporal local refinement. This scheme is studied in order to obtain a better understanding of more general multirate schemes. Issues of interest are local accuracy, propagation of interpolation errors and stability. Cases $\theta = 0$ (forward Euler), $\theta = 1$ (backward Euler), for which the θ-method is of first order, and $\theta = \frac{1}{2}$, for which the θ-method is of second order, are often used in practice. Missing component values, required during the refinement step, are computed using linear or quadratic interpolation. Analysis of the scheme together with numerical experiments shows that the use of linear interpolation can lead to an order reduction for stiff problems.

Chapter 3. In the third chapter, we compare the asymptotic stability properties of two multirate strategies: *recursive refinement* strategy and *compound step* strategy. For simplicity, only one level of refinement is considered.

In the *recursive refinement* strategy, a tentative macro-step of size Δt_n is performed first. For those components, where the solution is not accurate enough, the computation is redone with two micro-steps of size $\frac{1}{2}\Delta t_n$. This strategy allows for automatic partitioning based on the error estimators. For example, the multirate time stepping strategy presented in Chapter 1 can be used.

In the *compound step* strategy [2, 55], the macro-step and the first micro-step are computed simultaneously. The integration is continued with the second micro-step. The values at the macro-step time level for the active components are calculated twice in the *recursive refinement* strategy: the first time during the global step and the second time during the refinement step. The *compound step* strategy avoids this extra work. However, the partitioning in slow and fast components has to be done for this strategy in advance, before solving the system.

The scalar Dahlquist test equation cannot be used for the stability analysis of multirate methods, since multirate methods are used for systems, which have at least one slow and one fast components. Instead, we consider a linear 2×2 system. For each strategy we present the asymptotic stability regions and compare the results. The considered multirate schemes use second-order Rosenbrock type methods as the main time integration method. The results are given for linear- and quadratic interpolation at the refinement interface. It is also shown that the results, obtained for the simple 2×2 case, give a good indication for stability properties of more general systems, such as the semi-discrete systems obtained from the spatial discretization of the heat equation and the advection equation.

Chapter 4. Numerous multirate methods were developed for solving stiff systems with different time scales, e.g. [3, 16, 44, 47, 55]. All these schemes are of order two at most. In Chapter 4 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. 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. Use of Rosenbrock methods for problems with stiff source terms can lead to order reduction. We present a strategy, which helps us to recover the order of consistency for stiff problems, and which does not affect the order of consistency for non-stiff problems.

Chapter 5. Conservation laws give rise to mildly stiff ODE systems, upon space discretization, for which explicit time stepping methods can be used. Multirate schemes for semi-discrete conservation laws that have appeared in the literature all seem to have one of the following defects: either local inconsistency or lack of the mass conservation. In this chapter these two defects are discussed for one-dimensional conservation laws. Particular attention is given to monotonicity properties of the multirate schemes, such as maximum principles and the total variation diminishing (TVD) property. The study of these properties is done within the framework of partitioned Runge-Kutta methods. A detailed analysis of two multirate forward Euler schemes, proposed by Osher & Sanders [37] and Tang & Warnecke [54], is presented. Multirate schemes based on a standard

second-order two-stage Runge-Kutta method are also considered.

Chapter 1

A multirate time stepping strategy for stiff ordinary differential equations

To solve ODE systems with different time scales which are localized over the components, multirate time stepping is examined. In this chapter we introduce a self-adjusting multirate time stepping strategy, in which the step size for a particular component is determined by its own local temporal variation, instead of using a single step size for the whole system. We primarily consider implicit time stepping methods, suitable for stiff or mildly stiff ODEs. Numerical results with our multirate strategy are presented for several test problems. Comparisons with the corresponding single-rate schemes show that substantial gains in computational work and CPU times can be obtained.

1.1 Introduction

Standard single-rate time integration methods for ODEs work with time steps that are varying in time but constant over the components. There are, however, many problems of practical interest where the temporal variations have different time scales for different sets of the components. To exploit these local time scale variations, one needs multirate methods that use different, local time steps over the components.

In this chapter we will consider a simple multirate approach for system of ODEs

$$w'(t) = F\left(t, w(t)\right) , \quad w(0) = w_0 , \tag{1.1}$$

with given initial value in $w_0 \in \mathbb{R}^m$. The approximations at the global time levels t_n will be denoted by w_n.

Our multirate approach is based on local temporal error estimation. Given a global time step $\Delta t_n = t_n - t_{n-1}$, 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 $\frac{1}{2}\Delta t_n$. This refinement stage may require values at the intermediate

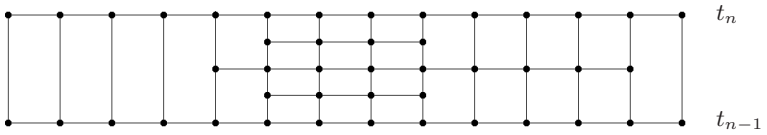


FIGURE 1.1: Multirate time stepping for a time slab $[t_{n-1}, t_n]$.

time level of components that are not refined. These values can be obtained by interpolation or by a 'dense output' formula. The refinement is continued with local steps $2^{-l}\Delta t_n$, until the error estimator is below a prescribed tolerance for all components. Schematically, with components horizontally and time vertically, the multirate time stepping is displayed in Figure 1.1. Small time steps will be used for the more active components and larger ones for the less active components.

The intervals $[t_{n-1}, t_n]$ are called time slabs. After each completed time slab the solutions are synchronized. In our approach, these time slabs are automatically generated, similar as in the single-rate approach, but without imposing temporal accuracy constraints on all components of (1.1).

An important issue in our strategy will be to determine the size of the time slabs. These could be taken large with many levels of refinements, or small with few refinements. A decision will be made based on an estimate of the number of components at which the solution needs to be calculated, including the overhead due to repeated computations in refined sets.

The problems (1.1) in this chapter are assumed to be stiff or mildly stiff. As basic integration method we will use a simple one-step Rosenbrock method. The presented strategy can be used with other methods as well, but for multistep methods additional interpolations of past values will be required in the refinement steps.

The chapter is organized as follows. In Section 1.2 we will briefly discuss related work on multirate schemes and introduce the Rosenbrock method that will be used as our basic numerical integration method. In Section 1.3 the multirate time stepping is described in detail, together with the time slab strategies. The performance of the schemes is discussed in Section 1.4 by means of several numerical experiments. Finally, Section 1.5 contains the conclusions and an outlook on further work.

1.2 Background material and preliminaries

1.2.1 Related work

The first descriptions of automatic multirate schemes were given by Gear and Wells [14] for linear multistep methods. As noted before, with multistep methods interpolations of past values will be needed in general in the temporal refinement stages.

In Günther, Kværnø and Rentrop [16] a multirate scheme was introduced which is based on partitioned Runge-Kutta methods with coupling between active and latent components performed by interpolation and extrapolation of state variables. In particular, they introduced the notion of a compound step in which the macro-step (for latent components) and the first micro-step (for the active components) are computed simultaneously. The partitioning into slow (latent) and fast (active) components is done in advance before solving the problem, based on knowledge of the ODE system to be solved (in their applications these where electrical circuits). A related scheme, based on Rosenbrock or ROW methods, was studied by Bartel and Günther [3]; this will be further discussed in Section 1.4.3. Some stability results for simplified versions of these schemes, applied to systems of two linear equations, with one fast and one slow component, have been presented in Kværnø [31].

An algorithm based on finite elements was proposed by Logg [34, 35]. In a single-rate approach such schemes are computationally akin to fully implicit Runge-Kutta methods. In the multirate approach this leads to very complicated implicit relations, which are difficult to solve. Additional remarks on the strategy used for this scheme can be found in Section 1.3.3.

Finally we mention that multirate schemes for explicit methods and non-stiff problems have been examined by Engstler and Lubich [10, 11]. In the first paper extrapolation is used, and in their strategy the partitioning into different levels of slow to fast components is obtained automatically during the extrapolation process. This approach looks quite promising, but for stiff problems and implicit methods the necessary asymptotic expansions seem difficult to obtain.

1.2.2 The Rosenbrock ROS2 method

Our multirate strategy is designed for one-step methods. In this chapter we will use the two-stage second-order Rosenbrock ROS2 method [27] as our basic numerical integration method. To proceed from t_{n-1} to a new time level $t_n = t_{n-1} + \tau$, the method calculates

$$\begin{aligned} w_n &= w_{n-1} + \tfrac{3}{2}\bar{k}_1 + \tfrac{1}{2}\bar{k}_2\,, \\ \big(I - \gamma\tau J\big)\bar{k}_1 &= \tau F(t_{n-1}, w_{n-1}) + \gamma\tau^2 F_t(t_{n-1}, w_{n-1})\,, \\ \big(I - \gamma\tau J\big)\bar{k}_2 &= \tau F(t_n, w_{n-1} + \bar{k}_1) - \gamma\tau^2 F_t(t_{n-1}, w_{n-1}) - 2\bar{k}_1\,, \end{aligned} \tag{1.2}$$

where $J \approx F_w(t_{n-1}, w_{n-1})$. The method is linearly implicit: to compute the internal vectors $\bar{k}_1$ and $\bar{k}_2$, a system of linear algebraic equations is to be solved. Method (1.2) is of order two for any choice of the parameter γ and for any choice of the matrix J. Furthermore, the method is A-stable for $\gamma \geq \frac{1}{4}$ and it is L-stable if $\gamma = 1 \pm \frac{1}{2}\sqrt{2}$. In this chapter we use L-stability with $\gamma = 1 - \frac{1}{2}\sqrt{2}$, since this gives smaller error coefficients in the local truncation error than the value $\gamma = 1 + \frac{1}{2}\sqrt{2}$. For the local error estimation within the variable step size

control we use the embedded first-order formula

$$\overline{w}_n = w_{n-1} + \bar{k}_1 \,. \tag{1.3}$$

We note that with our multirate approach, during the refinement step component values may be needed that are not included in this refinement. For example, components of $w(t)$ that are not refined may only be known in t_{n-1} and t_n; missing components will then be found by interpolation, and the F_t term in (1.2) will be approximated by

$$\widetilde{F}_t(t_{n-1}, w_{n-1}) \;=\; \frac{1}{\tau}\big(F(t_n, w_{n-1}) - F(t_{n-1}, w_{n-1})\big) \,. \tag{1.4}$$

This will not affect the order of the method. In all examples the exact Jacobian matrix $J = F_w(t_{n-1}, w_{n-1})$ will be used. For large practical problems a suitable approximation can be more efficient if that leads to more simple linear algebra systems.

The advantage of a Rosenbrock method is that only linear systems need to be solved. Implicit Runge-Kutta methods could also be used in our multirate approach, but then special attention should be given to the stopping criteria in Newton iterations. Making a large global time step with these methods might require many Newton iterations to get an iteration error smaller than a prescribed tolerance for the active components. But an accurate approximation is not needed there, because the numerical solution will be computed in the refinement steps. Therefore weighted norms should be used in the stopping criteria.

1.2.3 Variable step size control

Let us consider an attempted step from time t_{n-1} to $t_n = t_{n-1} + \tau_n$ with step size τ_n. Suppose this is done with two methods of order p and $p-1$, giving the numerical solutions w_n and $\overline{w}_n$, respectively. By comparing w_n with $\overline{w}_n$ we obtain an estimate for the local error,

$$E_n = \|w_n - \overline{w}_n\|_\infty \,. \tag{1.5}$$

Here the maximum norm is used because we aim at errors below the tolerance for all components.

Having the estimate E_n and a tolerance Tol specified by the user, two cases can occur: $E_n > Tol$ or $E_n \le Tol$. In the first case we decide to reject this time step and to redo it with a smaller step size τ_{new}, where we aim at $E_{new} = Tol$. In the second case we decide to accept the step and to continue the integration from t_n to t_{n+1}. In both cases we continue with a time step of size

$$\tau_{new} \;=\; \vartheta\, \tau_n \, \sqrt[p]{Tol/E_n} \,, \tag{1.6}$$

where the safety factor $\vartheta < 1$ serves to make the estimate conservative so as to avoid repeated rejections.

This form of variable step size selection is standard; see for example [19], [51]. We will use it in two ways in our multirate approach: to select the time slabs and to determine the components for which smaller step sizes are to be taken.

1.3 Multirate time stepping strategy

The time integration interval $[0, T]$ will be partitioned into synchronized time levels $0 = t_0 < t_1 < \ldots < t_N = T$. The length of the time slab $[t_{n-1}, t_n]$ will be denoted by Δt_n.

1.3.1 Strategy I : uniform treatment within time slabs

Processing of one time slab

Consider a single time slab $[t_{n-1}, t_n]$, as illustrated in Figure 1.1. Suppose that the approximation w_{n-1} at time t_{n-1} is known, and that we want to obtain an approximation w_n at the new time level. First we perform a single step with step size Δt_n and using an error estimator we determine the components for which the computation of the solution should be refined, that is, performed with a smaller time step. We refine for those components for which the estimated local error is larger than the prescribed tolerance *Tol*. This set of components is denoted as $\mathcal{J}_1$.

Refinement is done by doubling of the number of time steps for the selected set of components. So for all components in $\mathcal{J}_1$ we recalculate the solution using two steps of size $\frac{1}{2}\Delta t_n$. After this refinement phase we have the numerical solution for the set of components $\mathcal{J}_1$ at time levels $t_{n-1/2}$ and t_n. We then define $\mathcal{J}_2$ as the subset of components from $\mathcal{J}_1$ in which the estimated local error is still larger than the tolerance at either $t_{n-1/2}$ or t_n, and for all components from $\mathcal{J}_2$ we recalculate the solution using four time steps of size $\frac{1}{4}\Delta t_n$. This is repeated until the error estimator indicates that there is no need of smaller steps anymore. The processing of a time slab is then finished.

The interface, at the transition between the solutions obtained using different time step sizes, should be treated properly. For some components from the refinement set we will need solution values of components where we do not refine.

For example, in a first refinement step the solution is advanced for a part of the components using the halved time step $\frac{1}{2}\Delta t_n$. For the Rosenbrock method (1.2) this will require the values of the components at the time levels t_{n-1}, t_n and $t_{n-1/2}$. At time level t_{n-1} and t_n these values are available from the solution that has been computed with the coarse step Δt_n. At the intermediate time level $t_{n-1/2}$ we use interpolation based on the information available at t_{n-1} and t_n; this information consists of approximate solution values w_k and approximate derivative values $w'_k = F(t_k, w_k)$ for $k = n-1, n$.

In our tests, with the second-order method (1.2), we have examined linear interpolation based on w_{n-1} and w_n, and quadratic interpolation based on w_{n-1}, w'_{n-1} and w_n. For the numerical experiments presented in Section 1.4 both interpolations gave nearly identical results.

In general, the order of the interpolation should be related to the order of the time stepping method. With a basic integration method of order p, the error in one step will be $\sim \Delta t_n^{p+1}$. Interpolation with a q-th order polynomial will introduce an interpolation error $\sim \Delta t_n^{q+1}$ at the components in which we interpolate. Since we are interested in the errors in the maximum norm, the choice $q = p$ is natural. On the other hand, it was observed, also for higher-order methods, that taking $q = p-1$ often produces an order of accuracy equal to p for the whole scheme, due to damping and cancellation effects. A proper analysis for these effects is lacking at present.

Choosing the size of the time slabs

The size of the time slabs will be determined automatically while advancing in time. When we are done with the processing of the n-th time slab of size Δt_n, the size of the next time slab is taken as

$$\Delta t_{n+1} = 2^{s_{n+1}} \tau^*_{n+1}\,, \tag{1.7}$$

where s_{n+1} is the estimated (desired) number of levels of refinement for the $(n+1)$-st time slab, and τ^*_{n+1} is the optimal step size which would give us an estimated error smaller than the given tolerance if we were to use a single-rate approach for the next time step from t_n to $t_n+\tau^*_{n+1}$. Both s_{n+1} and τ^*_{n+1} will be estimated using information from the last time slab. In general, s_{n+1} may not coincide with the actual number of levels of refinement that will be taken; we will usually refine until the estimated error is smaller than the imposed tolerance. The estimations for s_{n+1} and τ^*_{n+1} will be discussed in the next subsections.

For the first time slab we use $s_1 = 0$, meaning that we would like to make a single time step with an estimated error less than the prescribed tolerance *Tol* at all components. The size of the first time slab Δt_1 is estimated using a small prescribed test step size τ_0 together with the step size control formula

$$\Delta t_1 \;=\; \vartheta\, \tau_0 \, \sqrt[p]{Tol/E_0}\,, \tag{1.8}$$

where the safety factor ϑ, the tolerance *Tol* and the order p of the method are as in (1.6), and E_0 is the maximum norm of the estimated local error for the test step from 0 to τ_0. In the numerical experiments presented in this chapter we use the ROS2 method ($p = 2$) with $\vartheta = 0.9$ and $\tau_0 = 10^{-4}$.

If the time integration is near an output point or the endpoint T, it should be verified whether $t_n + \Delta t_{n+1} > T$, and in that case we reset $\Delta t_{n+1} = T - t_n$.

In our implementation an additional check of the new time slab size Δt_{n+1} is made. This is to cover a situation where shortly after the last accepted time level t_n the solution suddenly becomes active. When after the global time step of size Δt_{n+1} has been performed it turns out that refinement is needed for each

component, then the size of the time slab is deemed too large. In that case a smaller size Δt_{new} will be selected by making a new estimate τ^*_{new} based on the newly available information and we also set $s_{new} = \max(0, s_{n+1} - 1)$. Such rejections will only occur in exceptional situations, with the sudden appearance of new active terms in the equations.

Estimation of τ^*_{n+1}

Using the information available from the n-th time slab we can estimate the value of τ^*_{n+1} for the next time slab. This is done using the standard step size control technique; the only difference is that for each component we use the information from the last available local time steps from the last time slab $[t_{n-1}, t_n]$. For example, in the time slab depicted in Figure 1.2, in order to estimate τ^*_{n+1}, we will use the information from the hatched areas where the last local time steps before t_n have been taken.

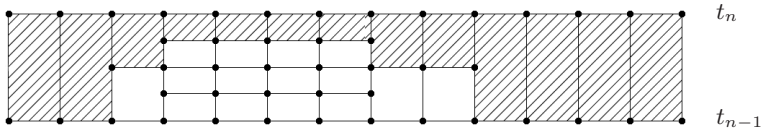


FIGURE 1.2: Time steps used for the estimation of τ^*_{n+1}.

After each level of refinement we know which components we already have refined (recall that for the k-th level of refinement this set of components is denoted by $\mathcal{J}_k$) and which components we ought to refine in the next level of refinement. Therefore, after the k-th level of refinement, for all components in $\mathcal{J}_k \setminus \mathcal{J}_{k+1}$, we estimate

$$\tau^{(k)}_{n+1} \;=\; \vartheta\, 2^{-k} \Delta t_n \sqrt[p]{Tol/E_k} \tag{1.9}$$

based on the local step sizes $2^{-k}\Delta t_n$ in the k-th level of refinement and on E_k, which is the maximum norm of the estimated error for the last time step at this level of refinement. The estimate in (1.9) represents the step size which would give us a local error smaller than the tolerance for all components from $\mathcal{J}_k \setminus \mathcal{J}_{k+1}$ if all is going well. The safety factor ϑ makes the estimate conservative.

After having finished with all levels of refinement we determine τ^*_{n+1} by

$$\tau^*_{n+1} \;=\; \min\left(\tau^{(0)}_{n+1}, \tau^{(1)}_{n+1}, \tau^{(2)}_{n+1}, \dots\right). \tag{1.10}$$

Expression (1.10) gives us an estimate of a step size with which we expect a local error smaller than the tolerance for all the components.

Estimation of s_{n+1}

The estimation of s_{n+1} will be based on the anticipated amount of work needed to cover a unit of time. The multirate approach will introduce component-time

points where the solution is computed several times, and this should be taken into account of course.

We suppose that the amount of work required for advancing one time step in m components is proportional to m^r with $r \geq 1$. In the experiments presented in this chapter we use the two-stage Rosenbrock method (1.2) as our time integration method. At each stage of this method one vector-function evaluation is done and one system of linear algebraic equations with a band matrix is solved. Therefore, in this chapter we can consider $r = 1$.

Suppose the n-th time slab has been processed using s_n levels of refinement, and that in the k-th level of refinement m_k components were refined, where $m_0 = m$. Since 2^k time steps were taken at this level of refinement to cover the time slab, the amount of work involved with the k-th level of refinement is $2^k m_k^r$. The amount of work per time unit for the processing of the entire time slab is therefore considered to be

$$C \;=\; \frac{1}{\Delta t_n}\big(m_0^r + 2m_1^r + \cdots + 2^{s_n} m_{s_n}^r\big). \tag{1.11}$$

In order to estimate the optimal amount of work per time unit we also study two hypothetical (virtual) computations for this last time slab. In the first case we consider what would have happened if we had taken the size of the time slab 2^l times smaller than Δt_n, and in the second case what would have happened if we had taken the size twice as large as Δt_n. In both cases we can estimate the amount of work per unit time, and this can be compared to the actual amount C. This information will then be used for the next time slab.

For the first hypothetical case, let us assume we go back to the n-th time slab and redo it with $\Delta t'_n = \frac{1}{2^l}\Delta t_n$, that is, 2^l times smaller than the actual Δt_n. Then we would start with a time step of size $\Delta t'_n$ on the whole spatial domain ($m_0 = m$ points). The number of components involved in the first refinement, with two steps of size $\frac{1}{2}\Delta t'_n = \frac{1}{2^{l+1}}\Delta t_n$, can be estimated to be m_{l+1}, because that was the number of components used in the actual computation with this time step. In the same way we can estimate that in the k-th level of refinement we would refine in m_{l+k} components and that $s_n - l$ levels of refinement would be used. Hence, the amount of work per time unit for this hypothetical case would be approximately

$$C' \;=\; \frac{1}{\Delta t'_n}\big(m_0^r + 2m_{l+1}^r + \cdots + 2^{s_n-l} m_{s_n}^r\big)\,. \tag{1.12}$$

If $C' < C$, we estimate that this hypothetical step would have given an improvement in the amount of work, compared to the actual computation that has been done.

Lemma 1.3.1 *Let $\rho = (\frac{1}{2})^{1/r}$. The value of C' in (1.12) attains its minimum for*

$$l_* \;=\; \max\{\, l \,:\, m_l > \rho\, m\}\,. \tag{1.13}$$

Proof. Denote the right-hand side of (1.12), with $\Delta t'_n = 2^{-l}\Delta t_n$, by C'_l. Then it is easily seen that

$$C'_{l-1} < C'_l \quad (\text{resp. } C'_{l-1} > C'_l) \quad \Longleftrightarrow \quad m_l < \rho m \quad (\text{resp. } m_l > \rho m)\,. \tag{1.14}$$

For the value l_* in (1.13) we have

$$m = m_0 \geq m_1 \geq \cdots \geq m_{l_*} > \rho m \geq m_{l_*+1} \geq \cdots \geq m_{s_n}\,.$$

It thus follows from (1.14) that

$$C'_0 > C'_1 > \cdots > C'_{l_*} \quad \text{and} \quad C'_{l_*} \leq C'_{l^*+1} \leq \cdots \leq C'_{s_n}\,,$$

which provides the proof of the lemma. □

If $l_* > 0$, then an improvement in amount of work per unit step could have been obtained if the n-th time slab had been done with fewer levels of refinement and a smaller size of the time slab. Therefore, for the $(n+1)$-st time slab we try to improve the performance by taking

$$s_{n+1} = s_n - l_*\,. \tag{1.15}$$

If $l_* = 0$, there was apparently no need to decrease the number of levels of refinement. But then more efficiency might be possible with a time slab of larger size (with more levels of refinement) than in the actual computation. This leads us to the second hypothetical case.

If the size of the n-th time slab had been two times larger than Δt_n, that is $\Delta t''_n = 2\Delta t_n$, then one time step of size $\Delta t''_n$ for all the components ($m_0 = m$ points) would have been performed, followed by refinement steps. Suppose that the first level of refinement would have involved m_* components. The second level of refinement then would take four time steps of size $\frac{1}{4}\Delta t''_n = \frac{1}{2}\Delta t_n$. In the processing of the original time slab of size Δt_n we needed time steps of this size in m_1 components. Therefore, it can be assumed that for the second level of refinement in the virtual step, refinement would also take place on m_1 components. Similarly, the k-th level of refinement can be assumed to involve m_{k-1} components. In total we would have $s_n + 1$ levels of refinement. The amount of work per time unit for this case would thus be approximately

$$C'' \;=\; \frac{1}{\Delta t''_n}\big(m_0^r + 2m_*^r + 2^2 m_1^r + \cdots + 2^{s_n+1} m_{s_n}^r\big). \tag{1.16}$$

In this case, taking the size of the time slab two times larger than Δt_n, would give us an expected improvement in work per time unit if $C > C''$, that is,

$$m_* \;<\; \rho\, m\,, \qquad \rho \;=\; \big(\frac{1}{2}\big)^{1/r}\,. \tag{1.17}$$

We still need an estimate for m_*. Let $v = w_n - \bar{w}_n$ be the difference between one step in the embedded Rosenbrock method (1.2), (1.3) computed in the n-th

time slab with step size Δt_n, and let $E_n = \|v\|_\infty$ be the norm of this estimated local error. Then $E_n \sim (\Delta t_n)^p$, with order $p = 2$ for the present Rosenbrock combination. In the first stage of our hypothetical step, starting from t_{n-1} with time step $2\Delta t_n$, we would expect an estimated local error of size $2^p E_n$. Consider the index set

$$\mathcal{I}_1 = \{\, i \,:\, |v_i| > 2^{-p}\, Tol \}\,, \tag{1.18}$$

where v_i is the i-th component of the vector v. Then m_* will be approximately equal to the number of elements $|\mathcal{I}_1|$ in this set. This estimate of m_* can be determined during the actual processing of the time slab without significant extra work. If

$$|\mathcal{I}_1| \,<\, \rho m\,, \tag{1.19}$$

then it is expected that a larger time slab with more refinement levels would have been more efficient. For the next time slab we then take $s_{n+1} = s_n + 1$. We note that a larger increase of refinement levels could be considered in a similar way, but it seems better to be conservative about this, because $s_{n+1} = s_n + 1$ will already lead (approximately) to a doubling of the size of the time slab (if $\tau^*_{n+1} \approx \tau^*_n$).

Summarizing, after having completed the n-th time slab with s_n levels of refinement, we choose for the next time slab

$$s_{n+1} \;=\; \begin{cases} s_n + 1 & \text{if (1.19) is satisfied}\,, \\ \\ s_n - l_* & \text{if (1.19) is not satisfied}\,, \end{cases} \tag{1.20}$$

where $l_* \geq 0$ is defined by (1.13). Together with (1.7) and (1.10) this determines the size Δt_{n+1} of the new time slab. The actual number of levels of refinements will be determined by the error estimations. The s_{n+1} in (1.20) is merely an indication for this. In our experiments the s_{n+1} was usually equal to the number of levels of refinements, but sometimes it was one more or one less.

1.3.2 Strategy II : recursive two-level approach

The time slab processing strategy presented in the above generally works very well, but in some cases a modification is desirable.

It may happen that the strategy takes very large time slabs with a large number of refinement levels. Then the smallest time steps are used throughout the entire time slab. Although this is only for a subset of components, it can be inefficient if the local temporal variation changes drastically inside this large time slab. Then the small time steps may be needed only in some part of the time slab $[t_{n-1}, t_n]$. In such a situation our strategy can be improved by applying the refinements not on the whole time slab but just for the required, smaller time intervals.

Let us consider a time slab $[t_{n-1}, t_n]$ with known approximation w_{n-1}. As before, we start with a single step $\Delta t_n = t_n - t_{n-1}$, and use the error estimator

to determine the components where we should refine in time. For those components the time slab is divided into two smaller sub-slabs with size $\frac{1}{2}\Delta t_n$. Next, each of these sub-slabs is processed separately, in a similar way as the initial 'global' time slab. This is a recursive processing strategy, which stops when the error estimator indicates that there is no need of further subslabs. A simple illustration for two levels of refinement is given in Figure 1.3.

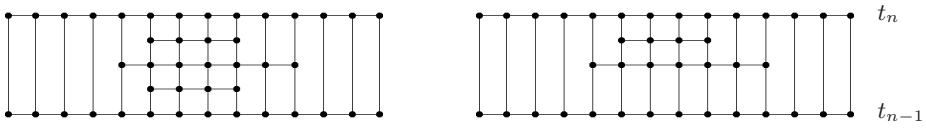


FIGURE 1.3: Example of a time slab created by the original strategy I (left) and the modified strategy II (right).

This modified time slab size processing strategy is considered in combination with a slightly modified time slab estimation. In the modified version the time slabs have a different structure; they are no longer uniform over the whole time slab. Therefore, not all the rationale from the previous time slab size estimation strategy can be used directly for the modified version.

The size of a time slab can still be determined using the same formula

$$\Delta t_{n+1} = 2^{s_{n+1}} \tau^*_{n+1}. \tag{1.21}$$

The value for τ^*_{n+1} can be determined using exactly the same procedure as in our original multirate strategy. The desired number of the levels of refinement s_{n+1} was determined on the basis of values of the number of components $m_0, m_1, \ldots, m_{s_n}$ in the levels of refinement for the n-th time slab. For the modified strategy these numbers of components are not constant anymore over the time slab. Still, for the new time slab we have as a first guess that the refinement will proceed uniformly as in the last local steps before t_n. Therefore, the estimations of the amount of work is done in the same way as before, but now with values of m_l based on the last available local steps before t_n. Using these values m_l we can determine the desired number of levels of refinement following the same procedure and rationale as in our original strategy. The size of the time slab obtained in this way is the optimal size which can be obtained based on the last information from the previous time slab.

1.3.3 Comparison to existing time slab strategies

Another time slab strategy has been presented by Jansson and Logg [28] for the multi-adaptive Galerkin time-stepping algorithm of Logg [34, 35]. In their strategy a time slab is created by first computing a desired time step for all components. The size of the time slab is then taken as $\Delta t = \theta \tau_{max}$ with τ_{max} the maximum over the desired time steps and $\theta \in (0, 1)$ a fixed parameter. The components are then partitioned into two sets. The components in the

group with large time steps are integrated with time step $\theta\Delta t$. The remaining components are processed by a recursive application of the same procedure.

In this multi-adaptive Galerkin approach, the resulting implicit systems for all refinements are solved simultaneously. This is the main difference with our approach. We first solve the coarse step, and then, successively, the refined steps. This leads to some overhead because in the refined regions the solution is computed repeatedly. On the other hand, with our approach the implicit systems are all relatively simple; basically the same as in a single-rate approach for (1.1) but with fewer points m_k in the refined steps. The dimension of the implicit systems in the approach of Logg will be very much larger than m, the number of components in (1.1), so these systems will be very hard to solve. For this reason a damped functional (fixed point) iteration is used in [28], but that can easily lead to a very large number of iterations per time slab.

In our case the size of a time slab is computed from the minimum time step over the components and an expected number of levels of refinement. In our strategy the sizes of the time slabs and the numbers of levels of refinement are automatically adjusted to get an optimal amount of work per time unit.

1.4 Numerical experiments

In this section we will present numerical results for several test problems. We consider the behavior of both our strategies: Multirate I (with uniform treatment within time slabs) and Multirate II (with the recursive two-level approach). The results are compared to the single-rate approach, also using the Rosenbrock pair (1.2) and (1.3).

As measure for the amount of work we consider primarily the number of components for which the solution is computed during the whole integration, where the fact that with our multirate approach some solution components will be computed several times at certain time levels is taken into account. For practical purposes the CPU time is more relevant, but this depends strongly on the programming language and environment. Some resulting computing times for a C-program will be discussed.

As mentioned before, the amount of work per step for m components in these experiments is estimated as m^r with $r = 1$. Tests with $r = 2$, which is obviously a wrong value here, produced quite similar results. In general, the choice of r will depend on the problem and linear algebra solver. The tests with $r = 2$ indicate that an optimal estimate for r is not critical for the performance of our multirate schemes.

One of the test problems is an ODE system from circuit analysis, the other two are obtained from partial differential equations (PDEs) by standard second-order central discretization of the spatial derivatives on fixed uniform grids (fourth-order central differences were also tried and the results were very similar). The resulting semi-discrete systems are simply considered as ODE test problems in these numerical experiments.

For the results reported here we used quadratic interpolation to obtain missing component values. Linear and cubic interpolation were also tried and the results were nearly identical; this simply indicates that the interpolation errors are not significant in these tests. Linear interpolation could potentially lower the order of accuracy, which is two for the ROS2 method, and therefore quadratic interpolation is our preferred interpolation here. As mentioned before, with a higher order basic time stepping method, also the order of interpolation should be increased. For a number of Runge-Kutta and Rosenbrock methods dense output formulas are available [19] which can also be considered.

The errors presented in the tables below are the maximum errors over the components at the output times T, with respect to a time-accurate ODE reference solution. The reference solutions have been computed by using very small tolerance values.

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

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

$$u_t \;=\; \epsilon u_{xx} + \gamma u^2(1-u), \tag{1.22}$$

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

$$u_x(0,t) \;=\; u_x(L,t) \;=\; 0\,, \qquad u(x,0) \;=\; \left(1+\mathrm{e}^{\lambda(x-1)}\right)^{-1}, \tag{1.23}$$

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 $\mathbb{R}^m$. An illustration of the semi-discrete solution at various times is given in Figure 1.4 with (spatial) components horizontally.

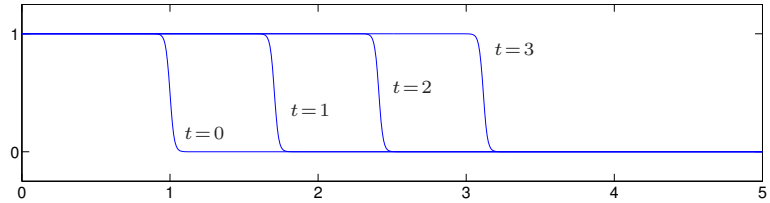


FIGURE 1.4: Traveling wave solution for problem (1.22)–(1.23) at various times.

In Table 1.1 the errors (in the maximum norm with respect to the reference ODE solution at time T) and the amount of work (number of space-time points for the integration interval $[0, T]$) 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 number of space-time points where the solution is computed is almost seven times larger. 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.

Measurements of CPU times (for a C-program) showed that for this problem the single-rate scheme was approximately four times more expensive than the multirate schemes. This factor four is less than the factor seven in space-time points; this is due to overhead with the multirate schemes for determining the time slabs and refinement regions.

The multirate strategy II (recursive two-level approach) works somewhat better for this problem than strategy I, in particular for the larger tolerances. In Figure 1.5 the space-time grid is shown on which the solution was calculated for strategy I with tolerance value $Tol = 2 \cdot 10^{-2}$. (With this large tolerance the structure of the grid is better visible than with small tolerances.) One nicely sees that the refinements move along with the steep gradient in the solution. From the more detailed picture (enlargement on part of the domain), it is seen that there is some redundancy in the fine level computations: in each time slab the fine level domains form a rectangle, and this is the reason why the strategy II is more efficient for this problem. Figure 1.6 shows the space-time grid for strategy II, again with $Tol = 2 \cdot 10^{-2}$.

TABLE 1.1: Errors and work amount for (semi-discrete) problem (1.22)–(1.23).

	Single-rate		Multirate I		Multirate II	
Tol	error	work	error	work	error	work
10^{-3}	$3.2 \cdot 10^{-3}$	818818	$3.4 \cdot 10^{-3}$	188138	$2.1 \cdot 10^{-3}$	124356
$5 \cdot 10^{-4}$	$1.9 \cdot 10^{-3}$	1128127	$1.9 \cdot 10^{-3}$	246962	$2.2 \cdot 10^{-3}$	149763
10^{-4}	$4.8 \cdot 10^{-4}$	2431429	$5.1 \cdot 10^{-4}$	411466	$5.4 \cdot 10^{-4}$	308685
$5 \cdot 10^{-5}$	$2.5 \cdot 10^{-4}$	3408405	$2.7 \cdot 10^{-4}$	550723	$2.7 \cdot 10^{-4}$	428549
10^{-5}	$5.3 \cdot 10^{-5}$	7528521	$5.5 \cdot 10^{-5}$	1153759	$5.7 \cdot 10^{-5}$	1064115

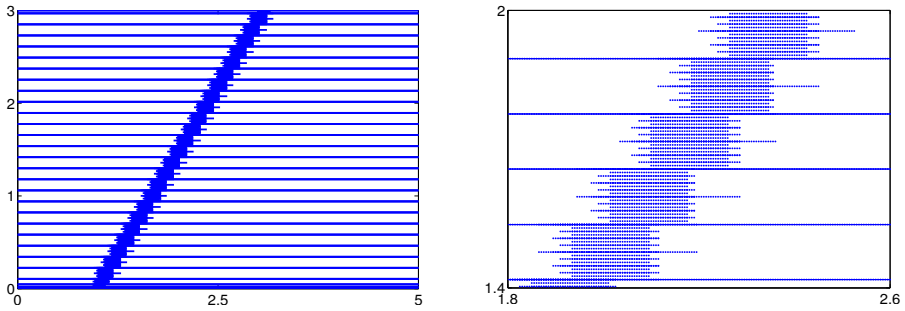


FIGURE 1.5: Space-time grid for problem (1.22)–(1.23) with strategy I. The right picture gives an enlargement for a part of the domain.

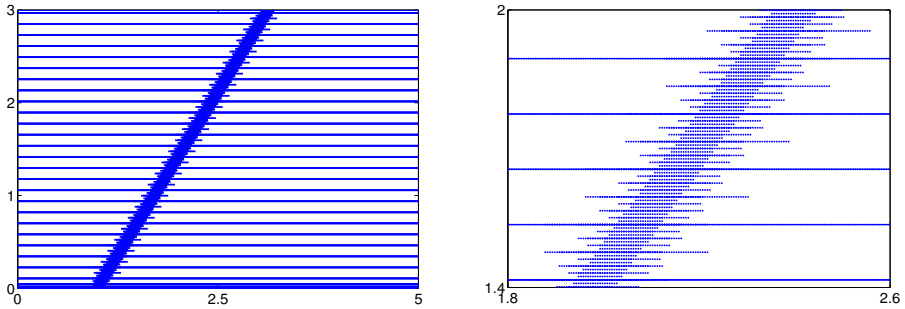


FIGURE 1.6: Space-time grid for problem (1.22)–(1.23) with strategy II. The right picture gives an enlargement for a part of the domain.

1.4.2 An ODE system obtained from semi-discretization: the Allen-Cahn equation

The second test consists of a semi-discrete version of the Allen-Cahn equation

$$u_t \;=\; \epsilon u_{xx} + u(1-u^2)\,, \tag{1.24}$$

for $t > 0$, $-1 < x < 2$, with initial- and boundary conditions

$$u_x(-1,t) = 0\,, \qquad u_x(2,t) = 0\,, \qquad u(x,0) = u_0(x)\,. \tag{1.25}$$

We take $\epsilon = 9 \cdot 10^{-4}$ and initial profile

$$u_0(x) = \begin{cases} \tanh((x+0.9)/(2\sqrt{\epsilon})) & \text{for } -1 < x < -0.7\,, \\ \tanh((0.2-x)/(2\sqrt{\epsilon})) & \text{for } -0.7 \le x < 0.28\,, \\ \tanh((x-0.36)/(2\sqrt{\epsilon})) & \text{for } 0.28 \le x < 0.4865\,, \\ \tanh((0.613-x)/(2\sqrt{\epsilon})) & \text{for } 0.4865 \le x < 0.7065\,, \\ \tanh((x-0.8)/(2\sqrt{\epsilon})) & \text{for } 0.7065 \le x < 2\,. \end{cases} \tag{1.26}$$

This problem is an extended version version of the bistable problem considered in [12].

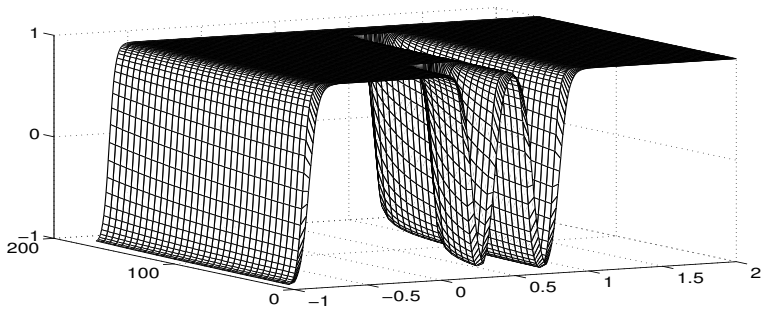


FIGURE 1.7: Evolution of the solution for problem (1.24)–(1.26).

For this problem we used a uniform space grid of 400 points with second-order central differences. Figure 1.7 shows a time-accurate numerical solution. The nonlinear reaction term in (1.24) has $u = 1$ and $u = -1$ as stable equilibrium states, whereas the zero solution is an unstable equilibrium. The solution of (1.24)–(1.26) starts with three 'wells', see Figure 1.7. The first well, on the left, persists during the integration interval. The second well is somewhat thinner than the others and it collapses at time $t \approx 41$, whereas the third well collapses at $t \approx 141$.

TABLE 1.2: Errors and work amount for (semi-discrete) problem (1.24)–(1.26).

	Single-rate		Multirate I		Multirate II	
Tol	error	work	error	work	error	work
$5 \cdot 10^{-4}$	$3.8 \cdot 10^{-3}$	102255	$3.0 \cdot 10^{-3}$	48342	$3.6 \cdot 10^{-3}$	36811
10^{-4}	$2.2 \cdot 10^{-3}$	217743	$1.5 \cdot 10^{-3}$	85241	$1.1 \cdot 10^{-3}$	66360
$5 \cdot 10^{-5}$	$1.2 \cdot 10^{-3}$	303958	$1.0 \cdot 10^{-3}$	107920	$1.3 \cdot 10^{-3}$	75653
10^{-5}	$2.8 \cdot 10^{-4}$	664858	$2.5 \cdot 10^{-4}$	257473	$2.6 \cdot 10^{-4}$	227554
$5 \cdot 10^{-6}$	$1.3 \cdot 10^{-4}$	935533	$1.1 \cdot 10^{-4}$	355627	$1.2 \cdot 10^{-4}$	324501

To test the performance of the schemes, the output was considered for $T = 142$. At this output point, the solution is still changing in the third well; for larger times the solution becomes steady-state and then all errors vanish. In Table 1.2 the errors (measured in the maximum norm with respect to the reference ODE solution) and the amount of work (number of space-time points) for different tolerances are presented. For this problem there is again a significant

improvement in work with the multirate schemes compared to the single-rate scheme.

Strategy II again behaves slightly better for this problem than strategy I. The error behavior of both multirate schemes is excellent: the errors are close to –or even smaller than– the errors of the single-rate scheme. As in the other tests, this shows that our multirate strategies behave very robustly.

In CPU times the factor gained with the multirate schemes, compared to the single-rate scheme, was a factor two approximately. As for the previous problem this is somewhat less than the factors for the number of space-time points due to overhead.

1.4.3 An Inverter Chain Problem

An inverter is an electrical sub-circuit which transforms a logical input signal to its negation. The inverter chain is a concatenation of several inverters, where the output of an inverter serves as input for the succeeding one. An inverter chain with an even number of inverters will delay a given input signal and will also provide some smoothing of the signal.

A detailed description of a mathematical model for an inverter chain is given in [2]. The model for m inverters consists of the equations

$$\begin{cases} w_1'(t) = U_{\rm op} - w_1(t) - \Upsilon g\big(u_{\rm in}(t), w_1(t)\big)\,, \\ w_j'(t) = U_{\rm op} - w_j(t) - \Upsilon g\big(w_{j-1}(t), w_j(t)\big)\,, \qquad j = 2, \ldots, m\,, \end{cases} \tag{1.27}$$

where

$$g(u,v) = \big(\max(u - U_{\rm thres}, 0)\big)^2 - \big(\max(u - v - U_{\rm thres}, 0)\big)^2. \tag{1.28}$$

The coefficient Υ serves as stiffness parameter. Following [2, 3], we solve the problem for a chain of $m = 500$ inverters with $\Upsilon = 100$, $U_{\rm thres} = 1$ and $U_{\rm op} = 5$. The initial condition is

$$w_j(0) = 6.247 \cdot 10^{-3} \ \text{ for } j \text{ even}, \quad w_j(0) = 5 \ \text{ for } j \text{ odd}. \tag{1.29}$$

The input signal is given by

$$u_{\rm in}(t) = \begin{cases} t - 5 & \text{for} \quad 5 \le t \le 10\,, \\ 5 & \text{for} \quad 10 \le t \le 15\,, \\ \frac{5}{2}(17 - t) & \text{for} \quad 15 \le t \le 17\,, \\ 0 & \text{otherwise.} \end{cases} \tag{1.30}$$

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

In Table 1.3 the errors at output time $T = 130$ (measured in the maximum norm with respect to an accurate reference solution) together with the amount

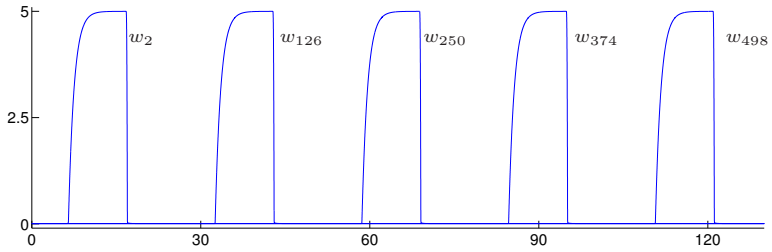


FIGURE 1.8: Solution components $w_j(t)$, $j = 2, 126, 250, 374, 498$, for problem (1.27)–(1.30).

of work and CPU times (in seconds) are presented for several tolerances for the single-rate scheme and the multirate strategy II. Similar as for the previous examples, the amount of work is taken as the number of components at which solutions are computed over the integration interval $[0, T]$; this is proportional to the number of scalar function evaluations (1.28).

It is seen from the table that for the prescribed tolerances we get roughly a factor 10 of improvement in work and a factor 6 improvement in CPU time with the multirate scheme, whereas for each given tolerance the errors of the multirate scheme are somewhat smaller than with the single-rate scheme.

In Figure 1.9 the component-time grid is shown on which the solution was calculated with tolerance value $tol = 5 \cdot 10^{-2}$. For this large tolerance the structure of the grid is better visible than for smaller tolerances, but still only every tenth global step is displayed in the left picture to make it more clear. Again it is seen that the refinements are properly adjusted to the steep gradients in the various components of the solution.

The same problem served as a numerical test for a multirate W-method in [3], where for each time slab a partitioning of the components in two classes, slow (latent) and fast (active), was used; the partitioning was based on a monitor function suited for this particular problem. Inside a time slab, all fast

TABLE 1.3: Errors and work amount for problem (1.27)–(1.29).

	Single-rate			Multirate II		
tol	error	work	CPU	error	work	CPU
$5 \cdot 10^{-4}$	$1.74 \cdot 10^{-1}$	28938500	19.75	$1.12 \cdot 10^{-1}$	3314690	3.74
$1 \cdot 10^{-4}$	$3.91 \cdot 10^{-2}$	62379000	42.64	$2.41 \cdot 10^{-2}$	4795878	6.36
$5 \cdot 10^{-5}$	$2.10 \cdot 10^{-2}$	87384000	59.72	$1.88 \cdot 10^{-2}$	6456558	8.81
$1 \cdot 10^{-5}$	$6.07 \cdot 10^{-3}$	193494000	132.32	$3.84 \cdot 10^{-3}$	17358472	21.65

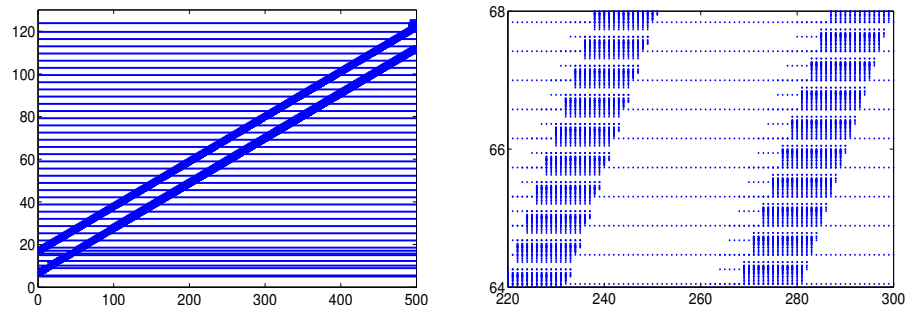


FIGURE 1.9: Component-time grid for problem (1.27)–(1.29) with strategy II. The right picture gives an enlargement for a part of the components and time interval.

components were then solved with the same small step sizes (micro-steps). In this way a factor 3.7 of improvement in work was obtained compared to the single-rate scheme. With our strategy this factor is much higher. This seems mainly due to the dynamic partitioning into several classes, together with the choices for the size of the time slabs and local time steps found by estimating the total amount of work.

Apart from the partitioning, the most important difference between [3] and our approach is the use of a 'compound step' in [3], whereby the slow components and the first (micro-) step for the fast components are solved simultaneously. Here extrapolation (from fast to slow components) and interpolation (from slow to fast components) is incorporated. In this way some of the overhead in our approach is avoided, because there are no coarse step values that are later overwritten, but such compound steps will become very complicated if the components are partitioned into more than two classes.

1.5 Conclusions

In this chapter we presented self-adjusting multirate time stepping strategies for stiff ODEs. The step size for a particular system component is determined by the local temporal variation of the solution, in contrast to the use of a single step size for the whole set of components as in the traditional (single-rate) methods. Numerical experiments confirmed that the efficiency of time integration methods can be significantly improved by using large time steps for inactive components, without sacrificing accuracy.

Although our two strategies produced results not too far apart, we do have a slight preference for the recursive two-level approach (strategy II) over the uniform treatment within time slabs (strategy I). Cases can be constructed with very large time slabs where strategy II will be much more efficient than strategy I.

Compared to the approaches in [3, 16] and [34, 35], our multirate approach

avoids the use of compound steps or very large implicit systems. On the other hand, there is some overhead with our approach, because in the refined component sets the solution is computed repeatedly. We do think, however, that for many problems simplicity will be preferable. Since the structure of the problems with the refined steps is the same as for the original problems, only on smaller component sets, linear algebra solvers suitable for the single-rate scheme can still be used.

As basic time stepping method, we used in this chapter a second-order Rosenbrock method with an embedded first-order method. Except for the interpolation, the multirate approach can be applied without adjustments to higher-order methods. Preliminary experiments with fourth-order Rosenbrock schemes are promising.

Acknowledgments

We would like to thank A. Verhoeven and E.J.W. ter Maten from Philips Research for suggesting the inverter chain test problem.

Chapter 2

Analysis of a multirate theta-method for stiff ODEs

This chapter contains a study of a simple multirate scheme, consisting of the θ-method with one level of temporal local refinement. Issues of interest are local accuracy, propagation of interpolation errors and stability. The theoretical results are illustrated by numerical experiments, including results for more levels of refinement with automatic partitioning.

2.1 Introduction

For large, stiff systems of ordinary differential equations (ODEs), some components may show a more active behavior than others. To solve such problems multirate time stepping schemes can be efficient. With such schemes different solution components can be integrated with different time steps. A multirate procedure with automatic partitioning and step size control was introduced and tested in Chapter 1. In the present chapter some theoretical issues are studied for a simplified situation. For this purpose we will consider the θ-method with one level of temporal refinement.

The systems of ODEs with given initial values in $\mathbb{R}^m$ are written as

$$w'(t) = F(t, w(t)) \,, \qquad w(0) = w_0 \,. \tag{2.1}$$

The numerical approximations to the exact ODE solution at the global time levels $t_n = n\tau$ will be denoted by w_n. For the step from t_{n-1} to t_n, we first compute a tentative approximation at the new time level. For those components for which an error estimator indicates that smaller steps would be needed, the computation is redone with halved step size $\frac{1}{2}\tau$. The result with the coarser time step will furnish data for this refined step by interpolation at the intermediate time level $t_{n-1/2} = \frac{1}{2}(t_{n-1} + t_n)$. This procedure then can be continued recursively with further refinements, but for the analysis here only the most simple case with one level of refinement will be considered.

We study this case to obtain a better understanding of more general multirate schemes. Particular attention will be given to the build-up of local errors

which will be composed of discretization errors of the θ-method and interpolation errors. For simplicity we consider the θ-method with a fixed global time step τ. The component set where the (halved) local time steps are taken is given by a diagonal projection J, with diagonal entries zero or one, where an entry one indicates that the component will be refined. Such J could be determined by some error estimator as in Chapter 1; it then will vary from step to step, so in general $J = J_n$.

Summarizing, the scheme reads as follows: first we take the tentative global step

$$\overline{w}_n \;=\; w_{n-1} + (1-\theta)\tau F(t_{n-1}, w_{n-1}) + \theta\tau F(t_n, \overline{w}_n)\,, \tag{2.2a}$$

from which we also obtain an approximation $\overline{w}_{n-1/2}$ at the intermediate time level $t_{n-1/2}$ by interpolation. Then we compute the local updates

$$\begin{aligned} w_{n-\frac{1}{2}} \;&=\; J_n\Big(w_{n-1} + \tfrac{1}{2}(1-\theta)\tau F(t_{n-1}, w_{n-1}) + \tfrac{1}{2}\theta\tau F(t_{n-\frac{1}{2}}, w_{n-\frac{1}{2}})\Big) \\ &\quad + (I - J_n)\overline{w}_{n-\frac{1}{2}}\,, \end{aligned} \tag{2.2b}$$

$$\begin{aligned} w_n \;&=\; J_n\Big(w_{n-\frac{1}{2}} + \tfrac{1}{2}(1-\theta)\tau F(t_{n-\frac{1}{2}}, w_{n-\frac{1}{2}}) + \tfrac{1}{2}\theta\tau F(t_n, w_n)\Big) \\ &\quad + (I - J_n)\overline{w}_n\,. \end{aligned} \tag{2.2c}$$

The θ-method is considered here as basic method since it represents the most simple Runge-Kutta method (and also linear multistep method). For stiff systems the cases $\theta = \frac{1}{2}$ (trapezoidal rule) and $\theta = 1$ (backward Euler) are of practical interest; for non-stiff systems we can also consider $\theta = 0$ (forward Euler). It is assumed in the following that $0 \le \theta \le 1$. For the interpolation we shall primarily consider linear interpolation

$$(I - J_n)\overline{w}_{n-\frac{1}{2}} = (I - J_n)\Big(\frac{1}{2}w_{n-1} + \frac{1}{2}\overline{w}_n\Big)\,. \tag{2.3}$$

However, we will see that this may affect the accuracy in case $\theta = \frac{1}{2}$, and therefore the quadratic interpolation formula

$$(I - J_n)\overline{w}_{n-\frac{1}{2}} = (I - J_n)\Big(\frac{3}{4}w_{n-1} + \frac{1}{4}\overline{w}_n + \frac{1}{4}\tau F(t_{n-1}, w_{n-1})\Big) \tag{2.4}$$

will be considered as well.

For this simple multirate scheme a detailed description of the error propagations will be derived for linear systems that may be stiff. Compared to the non-stiff case, it is not only stability that needs careful consideration, but also local discretization errors can be affected by stiffness. For example, it will be seen that for stiff systems the linear interpolation may give an $\mathcal{O}(\tau^2)$ contribution to the local error, whereas this contribution will always be $\mathcal{O}(\tau^3)$ for non-stiff systems.

Even though the multirate scheme considered in this chapter is quite simple, the stability analysis will turn out to be complicated. Some pertinent properties

for general linear systems can be derived, but to obtain detailed results we will also have to study linear test problems in $\mathbb{R}^2$.

Related stability results can be found in [16, 31, 50] for multirate schemes with a so-called compound step, where approximations $(I-J)w_n$ and $Jw_{n-1/2}$ are computed simultaneously. The above multirate approach (but then with more levels of refinement and with a two-stage Rosenbrock method as basic integrator) was considered in [47]. In this approach there is some overhead, because $J\overline{w}_n$ will not be directly used anymore. However, by computing the whole approximation $\overline{w}_n$, the structure of the implicit relations remains the same as for the corresponding single-rate scheme. Moreover, by using an embedded method, it is then relatively easy to make an automatic partitioning J_n based on local error estimations. For a detailed discussion and implementation issues we refer to [47]; some additional test results are presented in Section 2.5 of the present chapter.

The contents of this chapter is as follows. In Section 2.2 error recursions are derived that show how the global discretization errors for the multirate scheme are build up. Bounds for the local discretization errors are obtained in Section 2.3. Stability and contractivity properties of the multirate scheme are discussed in Section 2.4. In Section 2.5 some numerical test results are presented, both for the dual-rate scheme (2.2) used for the theoretical investigation and for an automatic multirate scheme, based on the trapezoidal rule, with local error estimation and variable time steps. Finally, Section 2.6 contains conclusions.

2.2 Error propagations

2.2.1 Preliminaries

For the analysis it will be assumed that the problem (2.1) is linear with constant coefficients,

$$w'(t) \;=\; Aw(t) + g(t) \tag{2.5}$$

with an $m \times m$ matrix $A = (a_{ij})$. In fact, to study the local truncation errors, the restriction to the linear constant-coefficient case is not necessary, but it gives a convenient compact notation. On the other hand, to obtain stability results we will also consider even more simple problems where $m = 2$.

For multirate schemes the aim is to have errors in active components of the same size as in components with larger timescales and less activity. Therefore the maximum norm is a natural norm to consider for analysis purposes. Stability of the multirate scheme will be considered under the assumption

$$a_{ii} + \sum_{j \neq i} |a_{ij}| \;\leq\; 0 \qquad \text{for } \; i = 1, \ldots, m\,. \tag{2.6}$$

In terms of logarithmic matrix norms this means $\mu_\infty(A) \leq 0$. It is well known, see [18, 27] for instance, that we then have $\|\exp(tA)\|_\infty \leq 1$ for all $t \geq 0$,

showing that initial perturbations are not amplified in the ODE system (2.5) itself.

Let in the following $Z = \tau A$. Furthermore, we denote the stability function of the θ-method by $R(z) = (1 + (1-\theta)z)/(1-\theta z)$. The corresponding matrix function is given by

$$R(Z) = (I - \theta Z)^{-1}(I + (1-\theta)Z) \tag{2.7}$$

where I is the identity matrix.

Let $e_n = w(t_n) - w_n$ be the global discretization error at time t_n. These global errors will satisfy a recursion of the form

$$e_n = S_n e_{n-1} + d_n\,. \tag{2.8}$$

This error recursion describes the amplification of existing errors, through S_n, and the appearance of new errors d_n during the step from t_{n-1} to t_n. These d_n are called the local discretization errors. The scheme is called consistent of order p if $\|d_n\| \le C\tau^{p+1}$. To have convergence of order p, that is, $\|e_n\| \le C\tau^p$ for all n, we will also need suitable bounds on the norms of (products of) the matrices S_n.

2.2.2 Error recursions

In this section recursions are derived for the global discretization errors e_n. The errors of the intermediate approximations are denoted in the same way as $\overline{e}_n = w(t_n) - \overline{w}_n$ and $e_{n+1/2} = w(t_{n+1/2}) - w_{n+1/2}$. The linear and quadratic interpolation formulas are covered by

$$(I - J_n)\overline{w}_{n-\frac{1}{2}} = (I - J_n)\Big(\tfrac{1}{2}w_{n-1} + \tfrac{1}{2}\overline{w}_n + \tfrac{1}{4}\gamma\big(w_{n-1} - \overline{w}_n + \tau F(t_{n-1}, w_{n-1})\big)\Big)$$

with $\gamma = 0$ for linear interpolation and $\gamma = 1$ for the quadratic case.

Inserting exact solution values into the scheme (2.2) gives residual errors in the various stages of the scheme, which are easily found by Taylor expansion. Subtraction of (2.2) then leads to the following error relations

$$\begin{aligned}
\overline{e}_n &= e_{n-1} + (1-\theta)Ze_{n-1} + \theta Z\overline{e}_n + \rho_{0,n}\,,\\
e_{n-\frac{1}{2}} &= J_n\Big(e_{n-1} + \frac{1}{2}(1-\theta)Ze_{n-1} + \frac{1}{2}\theta Ze_{n-\frac{1}{2}} + \rho_{1,n}\Big)\\
&\quad + (I - J_n)\Big(\frac{1}{2}e_{n-1} + \frac{1}{2}\overline{e}_n + \frac{1}{4}\gamma(e_{n-1} - \overline{e}_n + Ze_{n-1}) + \sigma_n\Big)\,,\\
e_n &= J_n\Big(e_{n-\frac{1}{2}} + \frac{1}{2}(1-\theta)Ze_{n-\frac{1}{2}} + \frac{1}{2}\theta Ze_n + \rho_{2,n}\Big) + (I - J_n)\overline{e}_n\,,
\end{aligned}$$

where the $\rho_{j,n}$ are local, residual errors caused by the underlying θ-method,

$$\begin{aligned}
\rho_{0,n} &= \Big(\frac{1}{2}-\theta\Big)\tau^2 w''(t_{n-\frac{1}{2}}) - \frac{1}{12}\tau^3 w'''(t_{n-\frac{1}{2}}) + \mathcal{O}(\tau^4)\,, \qquad (2.10a)\\
\rho_{1,n} &= \frac{1}{4}\Big(\frac{1}{2}-\theta\Big)\tau^2 w''(t_{n-\frac{1}{2}}) - \frac{1}{16}\Big(\frac{2}{3}-\theta\Big)\tau^3 w'''(t_{n-\frac{1}{2}}) + \mathcal{O}(\tau^4)\,,\\
\rho_{2,n} &= \frac{1}{4}\Big(\frac{1}{2}-\theta\Big)\tau^2 w''(t_{n-\frac{1}{2}}) + \frac{1}{16}\Big(\frac{1}{3}-\theta\Big)\tau^3 w'''(t_{n-\frac{1}{2}}) + \mathcal{O}(\tau^4)\,,
\end{aligned}$$

and

$$\sigma_n = \frac{1}{8}(\gamma-1)\tau^2 w''(t_{n-\frac{1}{2}}) - \frac{1}{48}\gamma\tau^3 w'''(t_{n-\frac{1}{2}}) + \mathcal{O}(\tau^4) \qquad (2.11)$$

is a residual error due to interpolation.

In the first stage of the scheme, with global step size τ, we thus obtain

$$\overline{e}_n = R(Z)e_{n-1} + (I-\theta Z)^{-1}\rho_{0,n}\,. \qquad (2.12)$$

At the first refined time level it follows that

$$\begin{aligned}
e_{n-\frac{1}{2}} &= (I-\tfrac{1}{2}\theta J_n Z)^{-1}\Big(J_n(I+\tfrac{1}{2}(1-\theta)Z) + (I-J_n)Q\Big)e_{n-1}\\
&+ (I-\tfrac{1}{2}\theta J_n Z)^{-1}\Big(J_n\rho_{1,n} + (I-J_n)\big(\sigma_n + (\tfrac{1}{2}-\tfrac{1}{4}\gamma)(I-\theta Z)^{-1}\rho_{0,n}\big)\Big)
\end{aligned} \qquad (2.13)$$

with interpolation matrix

$$Q = \frac{1}{2}I + \frac{1}{2}R(Z) + \frac{1}{4}\gamma\Big(I+Z-R(Z)\Big)\,. \qquad (2.14)$$

For the global discretization errors of the total scheme this finally leads to the error recursion (2.8) with amplification matrix

$$\begin{aligned}
S_n &= (I-\tfrac{1}{2}\theta J_n Z)^{-1}\Big(J_n R(\tfrac{1}{2}J_n Z)J_n(I+\tfrac{1}{2}(1-\theta)Z)\\
&\qquad + J_n R(\tfrac{1}{2}J_n Z)(I-J_n)Q + (I-J_n)R(Z)\Big)\,,
\end{aligned} \qquad (2.15)$$

and local discretization error

$$\begin{aligned}
d_n &= (I-\tfrac{1}{2}\theta J_n Z)^{-1}\Big(J_n R(\tfrac{1}{2}J_n Z)\big(J_n\rho_{1,n} + (I-J_n)\sigma_n\big) + J_n\rho_{2,n}\\
&\qquad + \big(I + (\tfrac{1}{2}-\tfrac{1}{4}\gamma)J_n R(\tfrac{1}{2}J_n Z)\big)(I-J_n)(I-\theta Z)^{-1}\rho_{0,n}\Big)\,.
\end{aligned} \qquad (2.16)$$

2.3 Local discretization errors

It is clear from (2.10), (2.11) that $\sigma_n = \mathcal{O}(\tau^3)$ if $\gamma = 1$ and $\rho_{j,n} = \mathcal{O}(\tau^3)$ if $\theta = \frac{1}{2}$. In other cases we only have $\mathcal{O}(\tau^2)$ bounds. Here the constants in the $\mathcal{O}(\tau^q)$ estimates are not affected by stiffness; they only depend on the

smoothness of the solution. To derive similar bounds for the local discretization errors it will be assumed that

$$\|R(\tau A)\|_\infty \le C\,, \qquad \|(I-\frac{1}{2}\tau\theta J_n A)^{-1}\|_\infty \le C\,, \tag{2.17}$$

with $C \ge 1$ a fixed constant. These assumptions are taken such that both cases $\theta = 0$ and $\theta > 0$ are covered. In fact, if $\theta > 0$ then (2.17) will be a consequence of (2.6), with C independent of τ, but for $\theta = 0$ it will impose a restriction on the step size.

Theorem 2.3.1 *Let $0 \le \theta \le 1$ and assume (2.17) holds. If $\theta = \frac{1}{2}$ and $\gamma = 1$, then $\|d_n\|_\infty = \mathcal{O}(\tau^3)$. Otherwise, we have $\|d_n\|_\infty = \mathcal{O}(\tau^2)$.*

Proof. For $\theta > 0$ assumption (2.17) implies

$$\|R(\tfrac{1}{2}J_nZ)\|_\infty \le \theta^{-1}(1-\theta) + \theta^{-1}\|(I-\tfrac{1}{2}\theta J_nZ)^{-1}\|_\infty \le \theta^{-1}(1-\theta+C)\,,$$

$$\|(I-\theta Z)^{-1}\|_\infty = \|(1-\theta)I + \theta R(Z)\|_\infty \le 1-\theta+\theta C\,,$$

whereas for $\theta = 0$, that is, $R(z) = 1+z$, we will have

$$\|R(\frac{1}{2}J_nZ)\|_\infty = \|(I-\frac{1}{2}J_n) + \frac{1}{2}J_n(I+Z)\|_\infty \le 1+\frac{1}{2}C\,.$$

Since $\|\rho_{j,n}\|_\infty = |\theta-\frac{1}{2}|\mathcal{O}(\tau^2) + \mathcal{O}(\tau^3)$ and $\|\sigma_n\|_\infty = |\gamma-1|\mathcal{O}(\tau^2) + \mathcal{O}(\tau^3)$, the required bounds thus follow from the local error expression (2.16). □

If $\theta \ne \frac{1}{2}$ this result cannot be improved in general, since the θ-method itself is then first-order consistent. The interesting question is whether we can have consistency of order two for $\theta = \frac{1}{2}$ with linear interpolation ($\gamma = 0$). The next result shows that will be valid if the coupling from the slow towards the more active components is bounded,

$$\|J_nA(I-J_n)\|_\infty \le K \tag{2.18}$$

with a moderate constant K. This will hold in particular for non-stiff problems.

Theorem 2.3.2 *Let $\theta = \frac{1}{2}$, $\gamma = 0$, and suppose that (2.17), (2.18) hold. Then we have the local error bound $\|d_n\|_\infty = \mathcal{O}(\tau^3)$.*

Proof. Since

$$J_nR(\frac{1}{2}J_nZ)(I-J_n) = J_n\Big(I + (I-\frac{1}{2}\theta J_nZ)^{-1}\frac{1}{2}J_nZ\Big)(I-J_n)\,,$$

it follows that

$$\|J_nR(\frac{1}{2}J_nZ)(I-J_n)\|_\infty \le \frac{1}{2}\|J_n(I-\frac{1}{2}\theta J_nZ)^{-1}\|_\infty\|J_nZ(I-J_n)\|_\infty \le \frac{1}{2}CK\tau\,.$$

Expression (2.16) thus leads directly to the proof. □

In case $\theta = \frac{1}{2}$ and $\gamma = 0$, but (2.18) is not satisfied with a moderate constant K, then the order of consistency will be less than two in general. For stiff systems, the order of convergence can be larger than the order of consistency, due to damping and cancellation effects (similar to [27, Lemma I.2.3] for Runge-Kutta methods), but we will see in Section 2.5 that for a simple example (semi-discrete heat equation) the scheme will not converge with order two.

2.4 Stability and contractivity

2.4.1 Contractivity with linear interpolation

Consider one step of (2.2) with $J_n = \mathrm{diag}(J_{ii})$. We denote by $\mathcal{I}_1 = \{i : J_{ii} = 0\}$ the index set where the step is not refined, and likewise $\mathcal{I}_2 = \{i : J_{ii} = 1\}$ stands for the index set where we do refine the step. For the multirate scheme we consider the time step restrictions

$$\begin{cases} |(1-\theta)\tau a_{ii}| \le 1 & \text{for } i \in \mathcal{I}_1\,, \\ |(1-\theta)\tau a_{ii}| \le 2 & \text{for } i \in \mathcal{I}_2\,. \end{cases} \tag{2.19}$$

Theorem 2.4.1 *Consider (2.15) with $0 \le \theta \le 1$ and $\gamma = 0$. Assume (2.6) and (2.19) are valid. Then $\|S_n\|_\infty \le 1$.*

Proof. From assumption (2.6) and the unconditional contractivity of the backward Euler method, see [19, 27] for instance, it follows that

$$\|(I-\theta Z)^{-1}\|_\infty \le 1\,, \qquad \|(I-\frac{1}{2}\theta J_n Z)^{-1}\|_\infty \le 1\,. \tag{2.20}$$

Moreover, the time step restriction (2.19) implies

$$\|(I-J_n)(I+(1-\theta)Z)\|_\infty \le 1\,, \qquad \|J_n(I+\frac{1}{2}(1-\theta)Z)\|_\infty \le 1\,.$$

We can write S_n as

$$S_n = (I-\tfrac{1}{2}\theta J_n Z)^{-1}\Big(J_n(I+\tfrac{1}{2}(1-\theta)Z)T_n + (I-J_n)R(Z)\Big)\,,$$
$$T_n = (I-\tfrac{1}{2}\theta J_n Z)^{-1}\Big(J_n(I+\tfrac{1}{2}(1-\theta)Z) + (I-J_n)Q\Big)\,,$$

where $Q = \frac{1}{2}(I+R(Z))$ for linear interpolation, see (2.13)–(2.15).

First consider the term

$$(I-J_n)R(Z) = (I-J_n)(I-\theta Z)^{-1}(I+(1-\theta)Z)\,.$$

Because $(I-\theta Z)^{-1}$ and $(I+(1-\theta)Z)$ commute we have

$$\|(I-J_n)R(Z)\|_\infty \le \|(I-J_n)(I+(1-\theta)Z)\|_\infty\,\|(I-\theta Z)^{-1}\|_\infty \le 1\,,$$

and consequently also

$$\|(I - J_n)Q\|_\infty \leq 1\,.$$

Using the fact that $\|J_nU + (I - J_n)V\|_\infty \leq 1$ for any two matrices $U, V \in \mathbb{R}^{m\times m}$ with $\|U\|_\infty \leq 1$, $\|V\|_\infty \leq 1$, it now easily follows that $\|T_n\|_\infty \leq 1$ and subsequently $\|S_n\|_\infty \leq 1$. □

The above conditions (2.19) for having $\|S_n\|_\infty \leq 1$ are sharp, as is seen from the following simple 3×3 example.

Example 2.4.1 Consider

$$A = \begin{pmatrix} -2 & -2 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & 0 \end{pmatrix}, \qquad J = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Then both restrictions in (2.19) reduce to

$$(1 - \theta)\tau \;\leq\; 1\,.$$

With $e = (1, 1, 1)^T$, it follows by some calculations that the second component of $S\,e$ is given by

$$(S\,e)_2 \;=\; -1 \;+\; 2\,\frac{1 - (1 - \theta)\tau}{1 + \theta\tau}\;.$$

It is now easily seen that $\|S\|_\infty > 1$ whenever (2.19) is not satisfied. □

So for the backward Euler case, $\theta = 1$, we will have unconditional contractivity; see [44] for a related (nonlinear) result for a backward Euler scheme with a compound step. Also for $\theta = 0$ (forward Euler) the result of Theorem 2.4.1 is entirely satisfactory; in fact, necessity of (2.19) is then already clear for diagonal matrices A. However, for $\theta = \frac{1}{2}$ (trapezoidal rule), the time step conditions in (2.19) are very strict. After all, the trapezoidal rule itself is A-stable.

The strict time steps for the trapezoidal rule are to some extent due to the insistence on contractivity, $\|S_n\| \leq 1$, rather than stability, where it is merely required that the error growth is moderate. From a practical point of view, having

$$\|S_nS_{n-1}\cdots S_2S_1\|_\infty \;\leq\; M \quad \text{for all } n \geq 1 \tag{2.21}$$

with some moderate constant M would be a sufficient stability condition. However, we will see below that for a standard linear example, arising from the heat equation, this will not be satisfied for the trapezoidal rule with linear interpolation if the step size τ is too large. This is due to the multirate procedure. The trapezoidal rule itself is stable in the maximum norm for this example (see e.g. [13]), and in the discrete L_2-norm it will even be contractive (see e.g. [18, 27]). The same heat equation example will also show that with quadratic interpolation stability can even be lost for $\theta = 1$.

2.4.2 Stability for fixed partitioning and non-stiff couplings

Consider $J_n = J$ fixed. Since the time step is also assumed to be constant, the amplification matrix S will then no longer depend on n either, so the stability condition (2.21) becomes the power boundedness condition $\|S^n\|_\infty \le M$. In the following we mostly restrict our attention to $\theta > 0$ and linear interpolation, $\gamma = 0$. Some remarks on quadratic interpolation are given near the end of this section.

For fixed J it can be assumed without loss of generality that

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \qquad J = \begin{pmatrix} O & \\ & I \end{pmatrix}. \tag{2.22}$$

This block partitioning can always be achieved by an index permutation. The same partitioning will be used for

$$Z = [Z_{ij}] = [\tau A_{ij}]\,, \quad R(Z) = [R(Z)_{ij}]\,, \quad Q = [Q_{ij}]\,, \quad S = [S_{ij}]\,.$$

Further we denote $U_{22} = (I - \frac{1}{2}\theta Z_{22})^{-1}$ for brevity. Then it is found by some calculations that the blocks of S are given by

$$\begin{cases} S_{11} = R(Z)_{11}\,, \quad S_{12} = R(Z)_{12}\,, \\ S_{21} = \frac{1}{2}(1-\theta)U_{22}R(\frac{1}{2}Z_{22})Z_{21} + \frac{1}{2}U_{22}^2 Z_{21}Q_{11} + \frac{1}{2}\theta U_{22}Z_{21}R(Z)_{11}\,, \\ S_{22} = R(\frac{1}{2}Z_{22})^2 + \frac{1}{2}U_{22}^2 Z_{21}Q_{12} + \frac{1}{2}\theta U_{22}Z_{21}R(Z)_{12}\,. \end{cases} \tag{2.23}$$

The actual form of the blocks $R(Z)_{ij}$ is somewhat complicated for general non-commuting A_{ij}, but if A is upper or lower block-triangular we obtain more simple expressions. Stability for those cases is considered under the following assumption on the diagonal blocks:

$$\|R(\tau A_{11})^n\|_\infty \le K_1 r_1^n\,, \quad \|R(\frac{1}{2}\tau A_{22})^{2n}\|_\infty \le K_2 r_2^n \quad \text{for } n \ge 1\,, \tag{2.24}$$

with $K_1, K_2 > 0$ and $0 \le r_1, r_2 \le 1$.

Theorem 2.4.2 *Assume $\theta > 0$, $\gamma = 0$, (2.6), (2.24), and let $r = \min(r_1, r_2)$. Furthermore, assume that either $A_{21} = 0$ or $A_{12} = 0$. Then there is a $K > 0$ such that*

$$\|S^n\|_\infty \;\le\; K \sum_{j=0}^{n} r^j \qquad \textit{for } n \ge 1\,.$$

Proof. We present the proof for the lower block-diagonal case $A_{12} = 0$. The proof for $A_{21} = 0$ is easier because most of the terms in (2.23) then cancel.

If $A_{12} = 0$, we find that $R(Z)_{12} = Q_{12} = 0$ and $R(Z)_{11} = R(Z_{11})$, which gives

$$S_{12} = 0\,, \qquad S_{11} = R(Z_{11})\,, \qquad S_{22} = R(\frac{1}{2}Z_{22})^2\,.$$

Moreover, from $S_{12} = 0$, it follows that

$$S^n = \begin{pmatrix} S_{11}^n & O \\ \sum_{j=1}^n S_{22}^{n-j} S_{21} S_{11}^{j-1} & S_{22}^n \end{pmatrix},$$

and hence

$$\|S^n\|_\infty \le \|S_{11}^n\|_\infty + \|S_{21}\|_\infty \sum_{j=1}^n \|S_{22}^{n-j}\|_\infty \|S_{11}^{j-1}\|_\infty + \|S_{22}^n\|_\infty .$$

It remains to show that $\|S_{21}\|_\infty$ is bounded. Let $U = [U_{ij}] = (I - \frac{1}{2}\theta JZ)^{-1}$. Then U_{22} is as above and $U_{21} = \frac{1}{2}\theta U_{22} Z_{21}$ in this lower block-diagonal case. Moreover, as seen in (2.20), assumption (2.6) implies $\|U\|_\infty \le 1$, and consequently also $\|U_{21}\|_\infty \le 1$, $\|U_{22}\|_\infty \le 1$. It thus follows that

$$U_{22} R(\frac{1}{2} Z_{22}) Z_{21} = \frac{\theta - 1}{\theta} U_{22} Z_{21} + \frac{1}{\theta} U_{22}^2 Z_{21}$$

can be bounded as well for $\theta > 0$. The same applies for the other terms in the expression (2.23) for S_{21}, where we note that $Q_{11} = \frac{1}{2}(I + R(Z_{11}))$ because of the linear interpolation. □

If $r < 1$ the theorem provides a stability result with $\|S^n\|_\infty \le K/(1-r)$ for all $n \ge 1$. If $r = 1$ it merely demonstrates weak stability $\|S^n\|_\infty \le Kn$ where a linear error growth is possible.

The above result for lower or upper block triangular matrices can be extended to non-stiff couplings by a perturbation argument, where we assume that A is not too far from a simpler matrix $\tilde{A}$ for which stability with the corresponding amplification matrix $\tilde{S}$ is known,

$$\|A - \tilde{A}\|_\infty \le L , \qquad \|\tilde{S}^n\|_\infty \le M \quad \text{for all } n \ge 1 . \tag{2.25}$$

Then stability of the scheme with the original amplification matrix S can be concluded on finite time intervals $0 \le t_n \le T$.

Theorem 2.4.3 *Suppose $\theta > 0$, $\gamma = 0$. Further assume that $\mu_\infty(\tilde{A}) \le 0$ and (2.25). Then there exist $C > 0$ and $\tau_* > 0$ (depending only on γ, L, M) such that*

$$\|S^n\|_\infty \le M \exp(CMt_n) \quad \textit{whenever } \tau \le \tau_* .$$

Proof. It is to be shown that $\|S - \tilde{S}\|_\infty \le C\tau$. Then the result follows from a standard perturbation argument; see for example [42, p. 58]. The estimate on $S - \tilde{S}$ requires some care.

We can decompose S as

$$S = VJ(I + \frac{1}{2}(1-\theta)Z) + V(I-J)Q + W , \tag{2.26}$$

where

$$V \,=\, (I - \tfrac{1}{2}\theta JZ)^{-1} JR(\tfrac{1}{2}JZ)\,, \quad W \,=\, (I - \tfrac{1}{2}\theta JZ)^{-1}(I-J)R(Z)\,. \tag{2.27}$$

For $\tilde{S}$ we consider the same form based on $\tilde{Z}$. Then

$$\begin{aligned} S - \tilde{S} \,=\, [V - \tilde{V}]J \,+\, \tfrac{1}{2}(1-\theta)[V-\tilde{V}]JZ \,+\, \tfrac{1}{2}(1-\theta)\tilde{V}J[Z-\tilde{Z}] \\ +\, [V-\tilde{V}](I-J)Q \,+\, \tilde{V}(I-J)[Q-\tilde{Q}] \,+\, [W-\tilde{W}]\,. \end{aligned} \tag{2.28}$$

Let us first consider $R(Z) - R(\tilde{Z})$. We have

$$\begin{aligned} R(Z) - R(\tilde{Z}) \,&=\, (I-\theta Z)^{-1}(Z-\tilde{Z})(I-\theta\tilde{Z})^{-1}\,, \\ (I-\theta Z)^{-1} \,&=\, \Big(I - \theta(I-\theta\tilde{Z})^{-1}(Z-\tilde{Z})\Big)^{-1}(I-\theta\tilde{Z})^{-1}\,. \end{aligned}$$

Since $\mu_\infty(\tilde{A}) \le 0$ we know that $\|(I-\theta\tilde{Z})^{-1}\|_\infty \le 1$. This leads to[1]

$$\|(I-\theta Z)^{-1}\|_\infty \,\le\, \frac{1}{1-\theta L\tau}\,, \qquad \|R(Z) - R(\tilde{Z})\|_\infty \,\le\, \frac{L\tau}{1-\theta L\tau} \,\le\, 2L\tau$$

provided that $\tau < 1/(2\theta L)$. The same applies to the perturbations for $R(\frac{1}{2}JZ)$. If we take $\tau_* = 1/(4\theta L)$ then these bounds are valid uniformly for $\tau \in (0, \tau_*]$.

The most difficult term to estimate in (2.28) is $[V - \tilde{V}]JZ$, because Z is not bounded by the assumptions. Denoting as before $U = (I - \frac{1}{2}\theta JZ)^{-1}$, we have

$$V - \tilde{V} \,=\, [U-\tilde{U}]JR(\frac{1}{2}JZ) + \tilde{U}J\big[R(\frac{1}{2}JZ) - R(\frac{1}{2}J\tilde{Z})\big]\,,$$

and hence

$$[V-\tilde{V}]JZ \,=\, \frac{1}{2}\theta\tilde{U}J[Z-\tilde{Z}]UJR(\frac{1}{2}JZ)JZ \,+\, \frac{1}{2}\tilde{U}J\tilde{U}J[Z-\tilde{Z}]UJZ\,.$$

Now, by noticing that

$$UJR(\frac{1}{2}JZ)JZ \,=\, \frac{\theta-1}{\theta}UJZ + \frac{1}{\theta}U^2JZ$$

and using the bounds for $\|U\|_\infty$ and $\|UJZ\|_\infty$ for $\tau \le \tau_*$, it follows that $\|[\tilde{V} - V]JZ\|_\infty$ can be bounded by $C\tau$. Estimation of the remaining terms in (2.28) proceeds in a similar way using the above estimates. □

The above perturbation result can be combined with Theorem 2.4.2 to obtain a stability result for non-stiff couplings, where either $\|A_{21}\|_\infty$ or $\|A_{12}\|_\infty$ is bounded by a moderate constant.

[1]Note that $\|X\| \le 1$ implies that $\|(I-XY)^{-1}\| \le (1 - \|Y\|)^{-1}$.

Remark 2.4.1 For $\theta = 0$ similar results can be derived if the assumptions $\mu_\infty(Z) \le 0$ or $\mu_\infty(\tilde{Z}) \le 0$ are replaced by appropriate boundedness assumptions. Results for $\theta > 0$ with quadratic interpolation require additional assumptions that are not satisfied in general for stiff systems. For example, in the proof of Theorem 2.4.2, an explicit Z_{11} terms then appears in Q_{11} in which case additional assumptions are needed to bound the term $U_{22}^2 Z_{21} Z_{11}$. For Theorem 2.4.3 it is similar. We will see in the next section that the stability properties of the multirate scheme are very poor indeed if quadratic interpolation is used, even if $\theta = 1$. □

2.4.3 Asymptotic stability for 2×2 test equations

In this section we present some detailed results on stability of the scheme (2.2) for the linear test equation (2.5) with real 2×2 matrices

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \qquad J = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.29}$$

We denote

$$\kappa = \frac{a_{11}}{a_{22}}, \qquad \beta = \frac{a_{12}a_{21}}{a_{11}a_{22}}. \tag{2.30}$$

By assumption (2.6) we have $\kappa \ge 0$ and $|\beta| \le 1$. 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 bounded by one in modulus. Similar stability considerations for 2×2 systems are found in [14, 16, 31, 43, 50, 55] for multirate schemes with a compound step.

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, the eigenvalues of S, which depend only on the determinant and trace of S, can be written as functions of three parameters: κ, β and z_{22}. This can be seen by elaborating (2.23) for this 2×2 case. Instead of $z_{22} \le 0$ we will use the quantity

$$\alpha = \frac{1 + \frac{1}{2}(1-\theta)z_{22}}{1 - \frac{1}{2}\theta z_{22}}, \tag{2.31}$$

which is bounded for $z_{22} \le 0$ and $\theta \ge \frac{1}{2}$.

The domains of asymptotic stability, where the spectral radius of S is bounded by one, are shown in the Figures 2.1–2.3 for $\theta = \frac{1}{2}, 1$ and linear or quadratic interpolation. We present these domains in the (α, β)-plane for three values of $\kappa = 10^j$, $j = 0, 1, 2$. Notice that $\alpha \in [-1, 1]$ if $\theta = \frac{1}{2}$ and $\alpha \in [0, 1]$ if $\theta = 1$. Generally the asymptotic stability domains are decreasing when κ is increased.

From Figure 2.1 it is seen that the combination of the trapezoidal rule and linear interpolation will be stable if $\beta \ge 0$, whereas for $\beta < 0$ the domain of

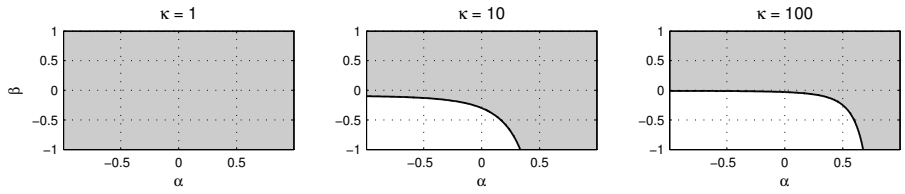


FIGURE 2.1: Asymptotic stability domains (gray areas) for the trapezoidal rule with linear interpolation, $\kappa = 1, 10, 100$.

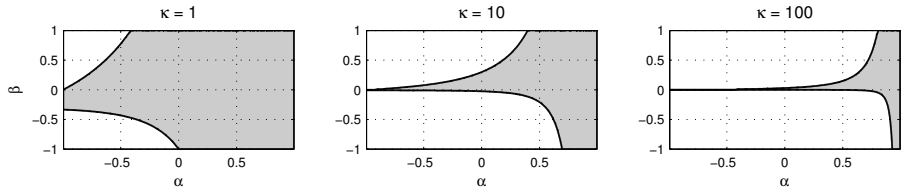


FIGURE 2.2: Asymptotic stability domains (gray areas) for the trapezoidal rule with quadratic interpolation, $\kappa = 1, 10, 100$.

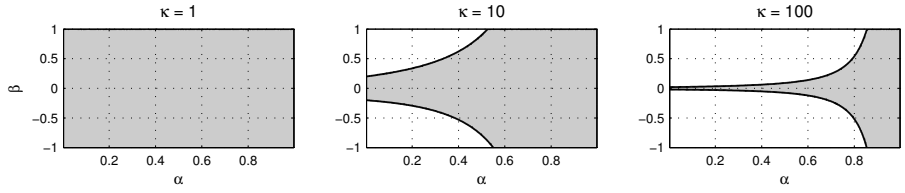


FIGURE 2.3: Asymptotic stability domains (gray areas) for backward Euler with quadratic interpolation, $\kappa = 1, 10, 100$.

instability increases when κ gets large. For the trapezoidal rule with quadratic interpolation, the scheme becomes unstable for large κ, unless $\beta = 0$. For both quadratic and linear interpolation, the limit case $\kappa \to 0$, with α, β fixed, gives stability of the scheme because then both z_{22} and z_{21} tend to zero as well.

As we already saw in Theorem 2.4.1, using the backward Euler method as underlying time integration method, the scheme will be stable with linear interpolation. However, as seen in Figure 2.3, the combination of backward Euler and quadratic interpolation is no longer stable when κ becomes large. Of course, in terms of accuracy it is for the backward Euler method not necessary to use quadratic interpolation, but the observed instability is of interest anyway.

Remark 2.4.2 Stability conditions based on eigenvalues of S are rather weak. If we have spectral radius $\rho(S) < 1$, then it is known that $S^n \to 0$ as $n \to \infty$, but

this does not guarantee that $\max_{n\geq 0} \|S^n\|_\infty$ is bounded by a moderate number because the bound may depend on τ and A. If $\rho(S) = 1$ is allowed, then even polynomial growth may occur. In our opinion, (2.29) is primarily a useful test equation for showing *instability* of certain schemes, such as the schemes with quadratic interpolation in this chapter. Demonstrating stability for (2.29) in some suitable norm is somewhat less relevant, because for an m-dimensional system with partitioning (2.22), the blocks A_{ij} may have complex eigenvalues, and, moreover, they will not commute in general. □

2.5 Numerical experiments

2.5.1 A linear parabolic example

As a test model we consider the parabolic equation

$$u_t + au_x \;=\; du_{xx} - cu + g(x,t)\,, \tag{2.32a}$$

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

$$u(x,0) = 0\,, \qquad u(-1,t) = 0\,, \qquad u(1,t) = 0\,. \tag{2.32b}$$

The constants and source term are taken as

$$a = 10\,, \quad d = 1\,, \quad c = 10^2\,, \quad g(x,t) = 10^3 \cos(\frac{1}{2}\pi x)^{100} \sin(\pi t)\,. \tag{2.32c}$$

The solution at the end time $T = 0.4$ is illustrated in Figure 2.4.

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 (2.5). 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.)

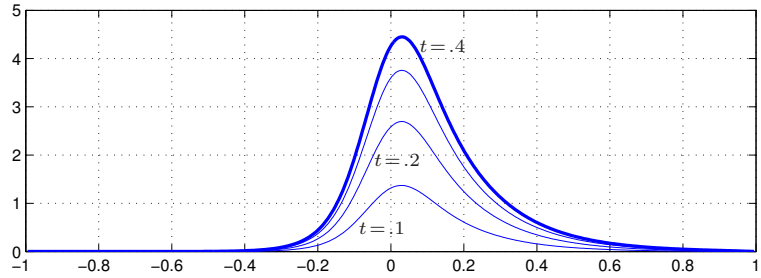


FIGURE 2.4: Solution for the parabolic test problem (2.32) at intermediate times $t = 0.1$, 0.2, 0.3 and the final time $t = T = 0.4$ (thick line).

Table 2.1 shows the discrete L_2-errors (scaled Euclidian norm) at $t = T$ with respect to a time-accurate ODE solution; the maximum errors were quite similar. The results are given for linear interpolation with the backward Euler method ($\theta = 1$) and the implicit trapezoidal rule ($\theta = \frac{1}{2}$), both with uniform, non-refined time steps $\tau = T/N$ and with locally refined steps $\tau/2$ on part of the spatial domain.

TABLE 2.1: Relative L_2-errors at $t = T$ versus N for the parabolic test problem. Results for the non-refined θ-method, $\theta = 1, \frac{1}{2}$, and for the scheme with one level of refinement and linear interpolation on the spatial region $-0.2 \le x_j \le 0.2$.

N	10	20	40	80	160
$\theta = 1$, non-ref.	$1.57 \cdot 10^{-3}$	$7.96 \cdot 10^{-4}$	$4.00 \cdot 10^{-4}$	$2.00 \cdot 10^{-4}$	$1.00 \cdot 10^{-4}$
$\theta = 1,\ \gamma = 0$	$1.21 \cdot 10^{-3}$	$5.93 \cdot 10^{-4}$	$2.86 \cdot 10^{-4}$	$1.37 \cdot 10^{-4}$	$6.55 \cdot 10^{-5}$
$\theta = \frac{1}{2}$, non-ref.	$1.81 \cdot 10^{-4}$	$3.76 \cdot 10^{-6}$	$8.12 \cdot 10^{-7}$	$2.03 \cdot 10^{-7}$	$5.07 \cdot 10^{-8}$
$\theta = \frac{1}{2},\ \gamma = 0$	$4.17 \cdot 10^{-4}$	$4.74 \cdot 10^{-5}$	$1.49 \cdot 10^{-5}$	$4.85 \cdot 10^{-6}$	$1.58 \cdot 10^{-6}$

The refinement region $-0.2 \le x_j \le 0.2$ was only chosen for test purposes; it is clear from Figure 2.4 that it is not a very good choice. Considering this fact, the results for $\theta = 1$ are satisfactory. However, for $\theta = \frac{1}{2}$ the errors with the local refinements are much larger than those for the non-refined scheme. This loss of accuracy is due to the linear interpolation, which lowers the order of consistency in this example.

Quadratic interpolation did give very large errors due to instabilities in this test, both for $\theta = 1$ (with errors in the range 10^2—10^{16}) and $\theta = \frac{1}{2}$ (errors in the range 10^7—10^{91}). In view of the unfavorable results that were found already for the 2×2 example in the previous section, this is not surprising anymore.

2.5.2 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

$$\begin{cases} w_1'(t) = U_{\text{op}} - w_1(t) - \Upsilon g\Big(u_{\text{in}}(t), w_1(t)\Big)\,, \\ w_j'(t) = U_{\text{op}} - w_j(t) - \Upsilon g\Big(w_{j-1}(t), w_j(t)\Big)\,, \qquad j = 2, \ldots, m\,, \end{cases} \tag{2.33a}$$

where

$$g(u, v) = \Big(\max(u - U_{\text{thres}}, 0)\Big)^2 - \Big(\max(u - v - U_{\text{thres}}, 0)\Big)^2. \tag{2.33b}$$

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

$$w_j(0) = 6.247 \cdot 10^{-3} \text{ for } j \text{ even}, \quad w_j(0) = 5 \text{ for } j \text{ odd}. \tag{2.33c}$$

The input signal is given by

$$u_{\text{in}}(t) = \begin{cases} t-5 & \text{for} \quad 5 \le t \le 10\,, \\ 5 & \text{for} \quad 10 \le t \le 15\,, \\ \frac{5}{2}(17-t) & \text{for} \quad 15 \le t \le 17\,, \\ 0 & \text{otherwise.} \end{cases} \tag{2.33d}$$

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

This problem is solved using the self-adjusting multirate time stepping strategy introduced in Chapter 1. Given a global time step $\Delta t_n = t_n - t_{n-1}$, 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 $\frac{1}{2}\Delta t_n$. The refinement is continued recursively with local steps $2^{-l}\Delta t_n$, until the error estimator is below a prescribed tolerance for all components. For details on the selection of the time step and number of refinement levels we refer to Chapter 1.

As the basic time integration method we use a linearized version of the trapezoidal rule,

$$w_n = w_{n-1} + \frac{1}{2}\tau\Big(F(t_{n-1}, w_{n-1}) + F(t_n, w_{n-1}) + A(w_n - w_{n-1})\Big) \tag{2.34}$$

where $A = \frac{\partial}{\partial w} F(t_n, w_{n-1})$. With this linearized trapezoidal rule nonlinear algebraic systems are avoided. To estimate the error of a step we compare the result with a step using the forward Euler method. It should be noted that the t_n argument is retained in the linearization (2.34). This done because the solution of this inverter chain problem has very steep temporal gradients, which are induced by earlier changes in the input function u_{in}. Further linearization, replacing $F(t_n, w_{n-1})$ in (2.34) by $F(t_{n-1}, w_{n-1}) + \tau \frac{\partial}{\partial t} F(t_{n-1}, w_{n-1})$, would give larger errors in this problem, because the forward Euler method also only uses information from time level t_{n-1}, so changes over the interval $[t_{n-1}, t_n]$ are then felt too late by the error estimator.

In Table 2.2 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. The results are given for linear interpolation at the coupling interface; quadratic interpolation gave similar results, without instabilities, in this example. As a measure for the amount of work we consider the total number of components at which solutions are computed over the complete integration interval $[0,T]$; this is proportional to the number of scalar function evaluations (2.33b). In addition, the CPU times are given.

TABLE 2.2: Absolute maximal errors, work amount and CPU times with different tolerances for the inverter chain problem.

	Single-rate			Multirate		
tol	error	work	CPU	error	work	CPU
$5 \cdot 10^{-4}$	$1.55 \cdot 10^{-1}$	32089000	20.12	$2.10 \cdot 10^{-1}$	4266674	4.06
$1 \cdot 10^{-4}$	$2.93 \cdot 10^{-2}$	70156000	44.06	$3.52 \cdot 10^{-2}$	7294108	7.52
$5 \cdot 10^{-5}$	$1.32 \cdot 10^{-2}$	98750000	61.97	$6.67 \cdot 10^{-3}$	9410734	9.94
$1 \cdot 10^{-5}$	$1.74 \cdot 10^{-3}$	219320500	137.76	$2.27 \cdot 10^{-3}$	26586200	23.14

It is seen from the table that for the prescribed tolerances we get roughly a factor 10 of improvement in work with the multirate scheme, compared to the standard single-rate method, whereas for each given tolerance the errors of the multirate scheme are of similar size as those of the single-rate scheme. In terms of CPU times we get a speed-up factor 6 approximately.

So for this test problem the multirate scheme with the (linearized) trapezoidal rule works well. There is no instability when using quadratic interpolation and there is no reduction in accuracy due to linear interpolation. It should be noted that this example is only mildly stiff, in contrast to the semi-discrete parabolic system in the first example.

Finally we note that the results in Table 2.2 are similar to those in Chapter 1 for a two-stage Rosenbrock method of order two. For practical problems that method seems preferable over the linearized trapezoidal rule (2.34), because the order of accuracy remains two if inexact Jacobians are used in the two-stage method. Moreover, the two-stage method allows an embedded (one-stage) method for error estimation with the same stability properties.

2.6 Conclusions

To obtain a better understanding of general multirate schemes, a simple scheme was studied in this chapter, with the θ-method as basic time integration method and with one level of refinement.

As seen from the local error bounds for the trapezoidal rule with linear interpolation ($\theta = \frac{1}{2}$, $\gamma = 0$), stiffness may lead to an order reduction where we obtain a lower order of consistency than for non-stiff problems.

A proper stability analysis is very difficult in general, even for the simple multirate scheme studied here. Detailed (numerical) results for very simple 2×2 cases are helpful to better understand possible instabilities for the schemes.

In spite of the lack of definitive theoretical results, multirate schemes can be efficient for problems with different levels of activities in the various components. The automatic partitioning strategy derived and tested in Chapter 1 (used in

this chapter for the inverter chain test problem with a linearized trapezoidal rule) provides in many cases of practical interest a significant speed-up compared to the corresponding single-rate scheme.

Finally we note that for higher-order Runge-Kutta or Rosenbrock schemes the class of possible interpolation formulas is larger than for the simple θ-method considered in this chapter, because then also internal stage values are available. For example, for the two-stage Rosenbrock method used in Chapter 1 preliminary tests have shown that there are interpolations of second-order consistency which are stable for the stiff test problems that were considered in this chapter. Extensions to methods of order larger than two are currently under investigation.

Chapter 3

Comparison of the asymptotic stability properties for two multirate strategies

This chapter contains a comparison of the asymptotic stability properties for two multirate strategies. For each strategy, the asymptotic stability regions are presented for a 2×2 test problem and the differences between the results are discussed. The considered multirate schemes use Rosenbrock type methods as the main time integration method and have one level of temporal local refinement. Some remarks on the relevance of the results for 2×2 test problems are presented.

3.1 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. In this chapter we will focus on two strategies: the *recursive refinement* strategy proposed in [25, 47] and the *compound step* strategy used in [2, 16, 53, 55]. We will analyze these multirate approaches for solving systems of ODEs

$$w'(t) = F(t, w(t)), \quad w(0) = w_0, \tag{3.1}$$

with $w_0 \in \mathbb{R}^m$.

In the *recursive refinement* strategy, given a global time step τ, a tentative approximation at the new time level is computed first. For those components, where the error estimator indicates that smaller steps would be needed, the computation is redone with a smaller time step $\frac{1}{2}\tau$. At this refinement stage, the values at the intermediate time levels of components which are not refined might be needed. These values can be calculated by using interpolation or a dense output formula. During a single global time step the refinement procedure

can be recursively continued until the local errors for all components are below a given tolerance, hence the name '*recursive*'. In our comparison in this chapter we consider only the most simple case with one level of refinement.

In the *compound step* strategy (sometimes also called *mixed compound-fast* [55]) the macro-step τ (for the slow components) and the first micro-step of a smaller size (for the active components) are computed simultaneously. Again, the values at the intermediate time levels of the slow components can be obtained by interpolation or dense output. This strategy may require values at the macro-step time level of the fast components. These values can be obtained by extrapolation. The integration is followed by a sequence of micro-steps for the fast components, until the time integration is synchronised with the slow components. In this chapter in the compound step strategy also only micro steps of size $\frac{1}{2}\tau$ are considered for the comparison with the recursive refinement strategy.

The values at the macro-step time level for the active components are calculated twice in the recursive refinement strategy, the first time during the global step and the second time during the refinement step. The compound step strategy avoids this extra work, however the partitioning in slow and fast components for this strategy has to be done in advance before solving the system. With the recursive refinement strategy, implicit relations of the same structure as with single-rate time stepping are obtained. The refinement step just leads to a system of smaller size. With the compound step strategy the compound step has a somewhat more complicated structure.

In this chapter we consider multirate schemes for systems with two levels of activity, slow and fast. It should be noted, however, that with the recursive refinement strategy it is easy to extend these schemes to multirate schemes with more levels of activity; for example, the multirate time stepping strategy presented in Chapter 1 can be used. With the compound step strategy handling more levels of activity is not easy.

In this chapter we study and compare asymptotic stability of these two multirate strategies for linear problems in $\mathbb{R}^2$. Our particular interest is to see how the extrapolation of the fast components affects the asymptotic stability of the scheme. A time integration method is called asymptotically stable if its amplification matrix S satisfies $||S^n|| \to 0$ when $n \to \infty$. A method is asymptotically stable if and only if all eigenvalues of S are inside the unit disk. Asymptotic stability does not guarantee stability, but it can help us with understanding the instability of some schemes. We also discuss the relevance of the results for the simple test equation in $\mathbb{R}^2$ for some interesting higher-dimensional systems.

The contents of this chapter is as follows. In Section 3.2 we introduce the Rosenbrock ROS1 and ROS2 methods which will be used as our basic numerical integration methods. In Section 3.3 we describe the 2×2 test problem for which the asymptotic stability domains are determined. The two multirate versions of ROS1 and ROS2 will be analysed in Sections 3.4 and 3.5. Some remarks on the relevance of the results for the 2×2 test problem are presented

in Section 3.6. Section 3.7 is devoted to a property of the eigenvalues of the partitioned Rosenbrock methods. Finally, Section 3.8 contains the conclusions.

3.2 Numerical integration methods ROS1 and ROS2

As the basic methods for the multirate schemes in this chapter we use two Rosenbrock methods [27]. The first method is a one-stage method, called in this chapter ROS1, which for non-autonomous systems $w'(t) = F(t, w(t))$ is given by

$$\begin{aligned} w_n &= w_{n-1} + k_1 \,, \\ \big(I - \gamma\tau J\big)k_1 &= \tau F(t_{n-1}, w_{n-1}) + \gamma\tau^2 F_t(t_{n-1}, w_{n-1}) \,, \end{aligned} \tag{3.2}$$

where w_n denotes the approximation to $w(t_n)$ and $J \approx F_w(t_{n-1}, w_{n-1})$. The method is of order two if $\gamma = \frac{1}{2}$. Otherwise the order is one. The method is A-stable for any $\gamma \geq \frac{1}{2}$ and L-stable for $\gamma = 1$. In this chapter we use $\gamma = \frac{1}{2}$.

The second method is the two stage second order method, to which we will refer to as ROS2,

$$\begin{aligned} w_n &= w_{n-1} + \tfrac{3}{2}\bar{k}_1 + \tfrac{1}{2}\bar{k}_2 \,, \\ \big(I - \gamma\tau J\big)\bar{k}_1 &= \tau F(t_{n-1}, w_{n-1}) + \gamma\tau^2 F_t(t_{n-1}, w_{n-1}) \,, \\ \big(I - \gamma\tau J\big)\bar{k}_2 &= \tau F(t_n, w_{n-1} + \bar{k}_1) - \gamma\tau^2 F_t(t_{n-1}, w_{n-1}) - 2\bar{k}_1 \,, \end{aligned} \tag{3.3}$$

where $J \approx F_w(t_{n-1}, w_{n-1})$. The method is also linearly implicit (to compute the internal vectors $\bar{k}_1$ and $\bar{k}_2$, a system of linear algebraic equations is to be solved), and it is of order two for any choice of the parameter γ and for any choice of the matrix J. Furthermore, the method is A-stable for $\gamma \geq \frac{1}{4}$ and it is L-stable if $\gamma = 1 \pm \frac{1}{2}\sqrt{2}$. In this chapter we use $\gamma = 1 - \frac{1}{2}\sqrt{2}$.

Other possible values of the parameter γ were also considered ($\gamma = 1$ for ROS1; $\gamma = \frac{1}{2}$ and $\gamma = 1 + \frac{1}{2}\sqrt{2}$ for ROS2). These values gave similar results and conclusions.

3.2.1 Interpolation and extrapolation

For given approximations $w_{n-1} \approx w(t_{n-1})$, $w_n \approx w(t_n)$, the multirate schemes will require an intermediate value $w_I(t_{n-\frac{1}{2}}) \approx w(t_{n-\frac{1}{2}})$. In [25] it was shown that for the multirate scheme based on the ROS1 method (with $\gamma = \frac{1}{2}$) and linear interpolation, stiffness may lead to an order reduction. For a special linear parabolic problem order 1.5 was obtained. Numerical experiments with the ROS2 method led to the same conclusion. Nevertheless, for many problems

order reduction will not be observed. Therefore, we consider in this chapter along with linear interpolation

$$w_I(t_{n-\frac{1}{2}}) = \frac{1}{2}\left(w_{n-1} + w_n\right) , \tag{3.4}$$

also forward quadratic interpolation

$$w_I(t_{n-\frac{1}{2}}) = \frac{3}{4}w_{n-1} + \frac{1}{4}w_n + \frac{1}{4}\tau F(t_{n-1}, w_{n-1}), \tag{3.5}$$

and backward quadratic interpolation

$$w_I(t_{n-\frac{1}{2}}) = \frac{1}{4}w_{n-1} + \frac{3}{4}w_n - \frac{1}{4}\tau F(t_n, w_n)\,. \tag{3.6}$$

With the ROS2 method we could also use what we call "embedded" quadratic interpolation, which uses the stages values of the method and avoids explicit evaluations of F:

$$w_I(t_{n-\frac{1}{2}}) = w_{n-1} + \frac{1}{8(1-2\gamma)}\left(5-12\gamma\right)\bar{k}_1 + \frac{1}{8(1-2\gamma)}\left(1-4\gamma\right)\bar{k}_2\,. \tag{3.7}$$

This interpolation mimics the quadratic interpolation based on $w(t_{n-1})$, $w(t_n)$ and $w'(t_{n-1}+\gamma\tau)$,

$$w_I(t_{n-\frac{1}{2}}) = \frac{1}{4(1-2\gamma)}\left((3-4\gamma)w_{n-1} + (1-4\gamma)w_n + \tau F(t_{n-1+\gamma}, w_{n-1+\gamma})\right)\,.$$

For linear problems and $\gamma = 1 \pm \frac{1}{2}\sqrt{2}$ the interpolation (3.7) coincides with (3.6). In the case of ROS1 with $\gamma = \frac{1}{2}$, backward quadratic interpolation is equivalent to the forward quadratic interpolation.

For the compound step strategy also extrapolation is needed: $w_E(t_n) \approx w(t_n)$. Again, we consider three types of extrapolation: linear

$$w_E(t_n) = 2w_{n-\frac{1}{2}} - w_{n-1}\,, \tag{3.8}$$

forward quadratic

$$w_E(t_n) = 4w_{n-\frac{1}{2}} - 3w_{n-1} - \tau F(t_{n-1}, w_{n-1})\,, \tag{3.9}$$

and backward quadratic

$$w_E(t_n) = w_{n-1} + \tau F(t_{n-\frac{1}{2}}, w_{n-\frac{1}{2}})\,. \tag{3.10}$$

Usually, for the compound step strategy, extra- and interpolations are done via internal stages.

3.3 The linear test problem in $\mathbb{R}^2$

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 consider a similar test problem, a linear 2×2 system

$$w'(t) = Aw(t)\,, \qquad w = \begin{pmatrix} u \\ v \end{pmatrix}\,, \qquad A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\,. \tag{3.11}$$

We denote

$$Z = \tau A\,, \quad z_{ij} = \tau a_{ij}\,. \tag{3.12}$$

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 denote

$$\kappa = \frac{a_{22}}{a_{11}}\,, \qquad \beta = \frac{a_{12}a_{21}}{a_{11}a_{22}}\,. \tag{3.13}$$

It will be assumed that

$$a_{11} < 0 \quad \text{and} \quad a_{22} < 0\,. \tag{3.14}$$

Then, 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

$$\beta < 1\,. \tag{3.15}$$

We can regard κ as a measure for the stiffness of the system, and β indicates the coupling between the fast and slow part of the system. For this two-dimensional test equation we will consider asymptotic stability whereby it is required that the eigenvalues of the amplification matrix of the multirate method are less than one in modulus. Instead of $z_{11} \leq 0$ and $\beta < 1$ it is convenient to use the quantities

$$\xi = \frac{z_{11}}{1 - z_{11}}, \quad \eta = \frac{\beta}{2 - \beta}\,, \tag{3.16}$$

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

3.4 Asymptotic stability for multirate ROS1

3.4.1 Recursive refinement strategy

In our recursive strategy, first we take the global step

$$\begin{aligned} \overline{w}_n &= w_{n-1} + k_1\,, \\ (I - \gamma Z)\, k_1 &= Z w_{n-1}\,, \end{aligned} \tag{3.17}$$

from which we also obtain an approximation $v_I(t_{n-\frac{1}{2}})$ for the second component at the intermediate time level $t_{n-\frac{1}{2}}$ by interpolation.

We continue with the first update step for the first component by solving the sub problem

$$u'(t) = a_{11}u(t) + a_{12}v_I(t)\,,$$

where the interpolant v_I is now considered as a time-dependent source term. We get

$$\begin{aligned} u_{n-\frac{1}{2}} &= u_{n-1} + \tilde{k}_1\,, \\ \left(1 - \frac{1}{2}\gamma z_{11}\right)\tilde{k}_1 &= \frac{1}{2}(z_{11}u_{n-1} + z_{12}v_{n-1}) + \frac{1}{4}\gamma z_{12}\tau v_I'(t_{n-1})\,, \end{aligned} \tag{3.18}$$

where the time derivative term is approximated by

$$\tau v_I'(t_{n-1}) = \overline{v}_n - v_{n-1} \tag{3.19}$$

without loosing the second order of the method.

At this point we have an numerical approximation of the solution at time $t_{n-\frac{1}{2}}$,

$$w_{n-\frac{1}{2}} = \begin{pmatrix} u_{n-\frac{1}{2}} \\ v_I(t_{n-\frac{1}{2}}) \end{pmatrix}. \tag{3.20}$$

We proceed with the second update step

$$\begin{aligned} u_n &= u_{n-\frac{1}{2}} + \hat{k}_1\,, \\ \left(1 - \frac{1}{2}\gamma z_{11}\right)\hat{k}_1 &= \frac{1}{2}(z_{11}u_{n-\frac{1}{2}} + z_{12}v_I(t_{n-\frac{1}{2}})) + \frac{1}{4}\gamma z_{12}\tau v_I'(t_{n-\frac{1}{2}})\,, \end{aligned} \tag{3.21}$$

where, again, we approximate

$$\tau v_I'(t_{n-\frac{1}{2}}) = \overline{v}_n - v_{n-1}\,, \tag{3.22}$$

without loosing the second order of the method. The final numerical value of the solution at time t_n is now given by

$$w_n = \begin{pmatrix} u_n \\ \overline{v}_n \end{pmatrix}. \tag{3.23}$$

3.4.2 Compound step strategy

In the compound step strategy, the first micro step for the first component

$$\begin{aligned} u_{n-\frac{1}{2}} &= u_{n-1} + k_1\,, \\ \left(1 - \frac{1}{2}\gamma z_{11}\right)k_1 &= \frac{1}{2}(z_{11}u_{n-1} + z_{12}v_{n-1}) + \frac{1}{4}\gamma z_{12}\tau v_I'(t_{n-1}) \end{aligned} \tag{3.24}$$

and the time step for the second component

$$\begin{aligned} v_n &= v_{n-1} + \hat{k}_1 \,, \\ (1 - \gamma z_{22})\, \hat{k}_1 &= (z_{21} u_{n-1} + z_{22} v_{n-1}) + \gamma z_{21} \tau u_I'(t_{n-1}) \end{aligned} \tag{3.25}$$

are computed at the same time. Then we continue with the second micro step for the first component

$$\begin{aligned} u_n &= u_{n-\frac{1}{2}} + \tilde{k}_1 \,, \\ \left(1 - \frac{1}{2}\gamma z_{11}\right) \tilde{k}_1 &= \frac{1}{2}(z_{11} u_{n-\frac{1}{2}} + z_{12} v_I(t_{n-\frac{1}{2}})) + \frac{1}{4}\gamma z_{12} \tau v_I'(t_{n-\frac{1}{2}}) \,. \end{aligned} \tag{3.26}$$

The time derivative terms are approximated by

$$\tau u_I'(t_{n-1}) = 2(u_{n-\frac{1}{2}} - u_{n-1}) \,, \tag{3.27}$$

$$\tau v_I'(t_{n-1}) = v_n - v_{n-1} \,, \tag{3.28}$$

$$\tau v_I'(t_{n-\frac{1}{2}}) = v_n - v_{n-1} \,. \tag{3.29}$$

Since these approximations are used for the $\tau^2 F_t$ term in (3.2), it follows that the order of the method does not change by (3.27)-(3.29). Relations

$$2(u_{n-\frac{1}{2}} - u_{n-1}) = 2k_1 \,,$$

$$v_n - v_{n-1} = \hat{k}_1 \,,$$

used for (3.27)-(3.29), reveal the joint computation of k_1 and $\hat{k}_1$ in (3.24)-(3.25).

3.4.3 Results

Both considered strategies can be written in the form of partitioned Rosenbrock methods (see for example [3]). Therefore the eigenvalues of the amplification matrix of the multirate schemes depend just on three parameters κ, η and ξ (see Section 3.7). The domains of asymptotic stability are shown in the Figures 3.1–3.4 for both strategies and all considered types of interpolation. We present these domains in the (ξ, η)-plane for three values of $\kappa = 10^j$, $j = 0, 1, 2$. We observe that for these multirate schemes the stability region decreases with the increasing of κ.

From Figure 3.1 and Figure 3.2 it is seen that the combination of ROS1 and linear interpolation is unconditionally stable for both multirate strategies if the coupling parameter $\eta \geq 0$. For the $\eta < 0$ case, both strategies have instability regions which increase when κ becomes large. In this case stability regions for the recursive refinement strategy are somehow larger than for the compound step strategy.

For the ROS1 with forward quadratic interpolation (Figure 3.3 and Figure 3.4), both multirate schemes become unstable for large κ, except the trivial

case $\eta = 0$. Both strategies have almost the same stability regions. The recursive refinement strategy has slightly larger stability area for $\eta > 0$. For $\eta < 0$ there exist a small set of points (close to $\xi = -0.8$) where the compound step strategy is asymptotically stable but the recursive refinement strategy is unstable. However, in general the recursive refinement strategy in the experiments in this section is slightly more stable.

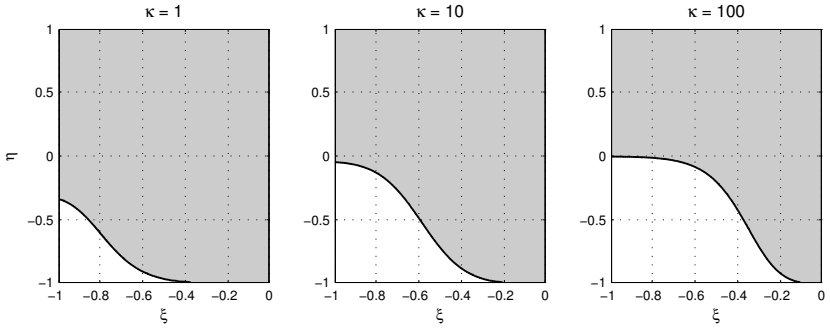


FIGURE 3.1: Recursive refinement, ROS1 with linear interpolation. Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

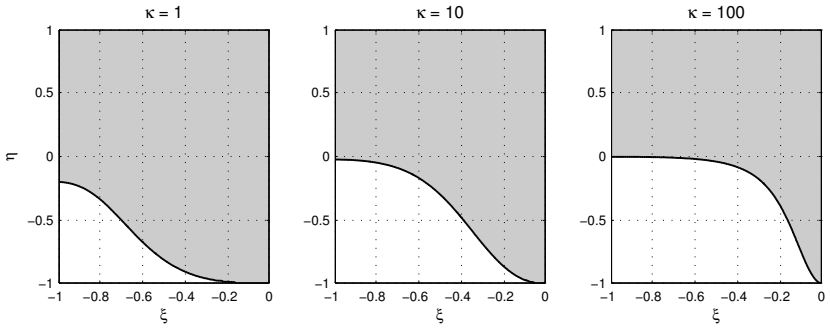


FIGURE 3.2: Compound step, ROS1 with linear interpolation. Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

The case $\eta \geq 0$ is relevant to the semi-discrete systems which are obtained by the central spatial discretization of the heat equation. The results obtained here suggest that the both strategies, based on ROS1 and linear interpolation, are stable for these semi-discrete systems. The results also show that for both strategies it is not possible to have an unconditionally stable second order multirate scheme based on ROS1. Using linear interpolation/extrapolation we get better stability properties, however we may lose one order due to stiffness (see the analysis in Chapter 2).

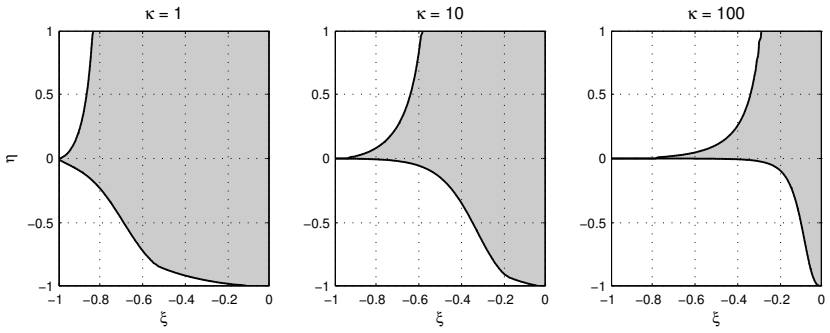


FIGURE 3.3: Recursive refinement, ROS1 with forward quadratic interpolation. Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

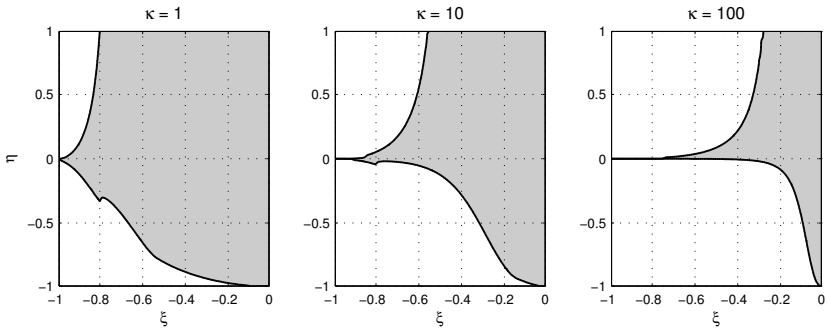


FIGURE 3.4: Compound step, ROS1 with forward quadratic interpolation. Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

3.5 Asymptotic stability for multirate ROS2

3.5.1 Recursive Refinement Strategy

In our recursive strategy, first we take the global step

$$\begin{aligned} \overline{w}_n &= w_{n-1} + \frac{3}{2}\bar{k}_1 + \frac{1}{2}\bar{k}_2\,, \\ (I-\gamma Z)\,\bar{k}_1 &= Z w_{n-1}\,, \\ (I-\gamma Z)\,\bar{k}_2 &= Z(w_{n-1}+\bar{k}_1) - 2\bar{k}_1\,, \end{aligned} \tag{3.30}$$

from which we also obtain an approximation $v_I(t_{n-\frac{1}{2}})$ for the second component at the intermediate time level $t_{n-\frac{1}{2}}$ by interpolation.

We continue with the first update step for the first component

$$\begin{aligned} u_{n-\frac{1}{2}} &= u_{n-1} + \frac{3}{2}\tilde{k}_1 + \frac{1}{2}\tilde{k}_2\,, \\ \left(1 - \frac{1}{2}\gamma z_{11}\right)\tilde{k}_1 &= \frac{1}{2}(z_{11}u_{n-1} + z_{12}v_{n-1}) + \frac{1}{4}\gamma z_{12}\tau v_I'(t_{n-1})\,, \\ \left(1 - \frac{1}{2}\gamma z_{11}\right)\tilde{k}_2 &= \frac{1}{2}(z_{11}(u_{n-1} + \tilde{k}_1) + z_{12}v_I(t_{n-\frac{1}{2}})) - \frac{1}{4}\gamma z_{12}\tau v_I'(t_{n-1}) - 2\tilde{k}_1\,, \end{aligned} \tag{3.31}$$

where the time derivative term is approximated with

$$\tau v_I'(t_{n-1}) = \overline{v}_n - v_{n-1}\,. \tag{3.32}$$

Since this approximation is used for the $\tau^2 F_t$ term in (3.3), it follows that the order of the method does not change by (3.32).

At this point we get the numerical approximation of the solution at time $t_{n-\frac{1}{2}}$

$$w_{n-\frac{1}{2}} = \begin{pmatrix} u_{n-\frac{1}{2}} \\ v_I(t_{n-\frac{1}{2}}) \end{pmatrix}. \tag{3.33}$$

We proceed further with the second update step

$$\begin{aligned} u_n &= u_{n-\frac{1}{2}} + \frac{3}{2}\hat{k}_1 + \frac{1}{2}\hat{k}_2\,, \\ \left(1 - \frac{1}{2}\gamma z_{11}\right)\hat{k}_1 &= \frac{1}{2}(z_{11}u_{n-\frac{1}{2}} + z_{12}v_I(t_{n-\frac{1}{2}})) + \frac{1}{4}\gamma z_{12}\tau v_I'(t_{n-\frac{1}{2}})\,, \\ \left(1 - \frac{1}{2}\gamma z_{11}\right)\hat{k}_2 &= \frac{1}{2}(z_{11}(u_{n-\frac{1}{2}} + \hat{k}_1) + z_{12}\overline{v}_n) - \frac{1}{4}\gamma z_{12}\tau v_I'(t_{n-\frac{1}{2}}) - 2\hat{k}_1\,, \end{aligned} \tag{3.34}$$

where, again, we approximate

$$\tau v_I'(t_{n-\frac{1}{2}}) = \overline{v}_n - v_{n-1}. \tag{3.35}$$

The final numerical value of the solution at time t_n is given by

$$w_n = \begin{pmatrix} u_n \\ \overline{v}_n \end{pmatrix}. \tag{3.36}$$

3.5.2 Compound step strategy

In the compound step strategy, the first micro step for the first component

$$\begin{aligned} u_{n-\frac{1}{2}} &= u_{n-1} + \frac{3}{2}\bar{k}_1 + \frac{1}{2}\bar{k}_2\,, \\ \left(1 - \frac{1}{2}\gamma z_{11}\right)\bar{k}_1 &= \frac{1}{2}(z_{11}u_{n-1} + z_{12}v_{n-1}) + \frac{1}{4}\gamma z_{12}\tau v_I'(t_{n-1})\,, \\ \left(1 - \frac{1}{2}\gamma z_{11}\right)\bar{k}_2 &= \frac{1}{2}(z_{11}(u_{n-1} + \bar{k}_1) + z_{12}v_I(t_{n-\frac{1}{2}})) - \frac{1}{4}\gamma z_{12}\tau v_I'(t_{n-1}) - 2\bar{k}_1 \end{aligned} \tag{3.37}$$

and the time step for the second component

$$\begin{aligned} v_n &= v_{n-1} + \frac{3}{2}\hat{k}_1 + \frac{1}{2}\hat{k}_2\,, \\ (1-\gamma z_{22})\,\hat{k}_1 &= (z_{21}u_{n-1} + z_{22}v_{n-1}) + \gamma z_{21}\tau u_I'(t_{n-1})\,, \\ (1-\gamma z_{22})\,\hat{k}_2 &= (z_{21}u_E(t_n) + z_{22}(v_{n-1}+\hat{k}_1)) - \gamma z_{21}\tau u_I'(t_{n-1}) - 2\hat{k}_1 \end{aligned} \tag{3.38}$$

are computed at the same time. Then we continue with the second micro step

$$\begin{aligned} u_n &= u_{n-\frac{1}{2}} + \frac{3}{2}\tilde{k}_1 + \frac{1}{2}\tilde{k}_2\,, \\ \left(1-\frac{1}{2}\gamma z_{11}\right)\tilde{k}_1 &= \frac{1}{2}(z_{11}u_{n-\frac{1}{2}} + z_{12}v_I(t_{n-\frac{1}{2}})) + \frac{1}{4}\gamma z_{12}\tau v_I'(t_{n-\frac{1}{2}})\,, \\ \left(1-\frac{1}{2}\gamma z_{11}\right)\tilde{k}_2 &= \frac{1}{2}(z_{11}(u_{n-\frac{1}{2}}+\tilde{k}_1) + z_{12}v_n) - \frac{1}{4}\gamma z_{12}\tau v_I'(t_{n-\frac{1}{2}}) - 2\tilde{k}_1. \end{aligned} \tag{3.39}$$

The time derivative terms are approximated by

$$\tau u_I'(t_{n-1}) = 2(u_{n-\frac{1}{2}} - u_{n-1})\,, \tag{3.40}$$

$$\tau v_I'(t_{n-1}) = v_n - v_{n-1}\,, \tag{3.41}$$

$$\tau v_I'(t_{n-\frac{1}{2}}) = v_n - v_{n-1}\,. \tag{3.42}$$

Again, these approximations will not affect the order of the method.

A multirate scheme based on a third-order Rosenbrock method and compound step strategy was considered in [3]. Due to stability constraints, instead of the third-order method the embedded second-order method was used for time stepping. Extra- and interpolations were done via internal stages.

3.5.3 Results

Again, both considered strategies can be written in the form of a partitioned Rosenbrock methods (for example by adding some artificial extra stages to the original method). Therefore the eigenvalues of the amplification matrix of the multirate schemes will depend on three parameters κ, η and ξ (see Section 3.7).

The domains of asymptotic stability are shown in the Figures 3.5–3.10 for both strategies and all considered types of interpolation/extrapolation. We present these domains in the (ξ,η)-plane for three values of $\kappa = 10^j$, $j = 0, 1, 2$. From Figure 3.5 and Figure 3.6 it is seen that the combination of ROS2 and linear interpolation is unconditionally stable for both multirate strategies if $\eta \geq 0$. An instability region appears at η close to -1. The instability region for the recursive refinement strategy is smaller than for the compound step strategy.

For ROS2 with forward quadratic interpolation (Figures 3.7 and 3.8), both multirate schemes become unstable for large κ, unless $\eta = 0$. In this case the recursive refinement strategy has larger stability regions than the compound

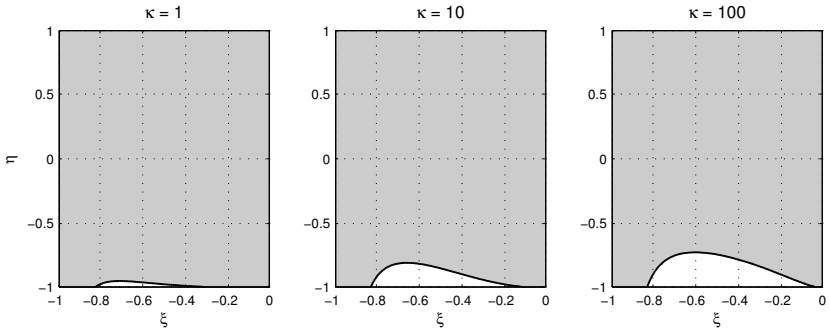


FIGURE 3.5: Recursive refinement, ROS2 with linear interpolation. Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

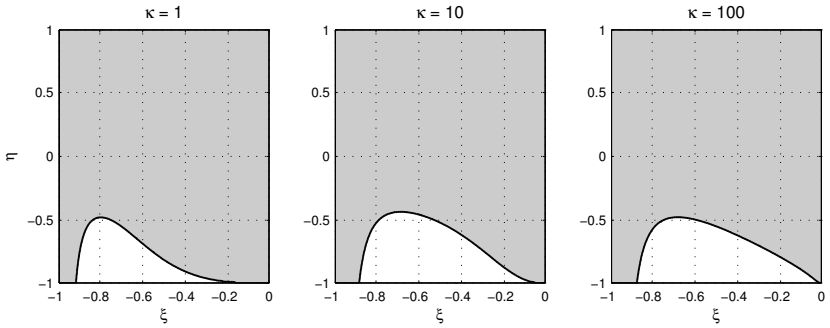


FIGURE 3.6: Compound step, ROS2 with linear interpolation. Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

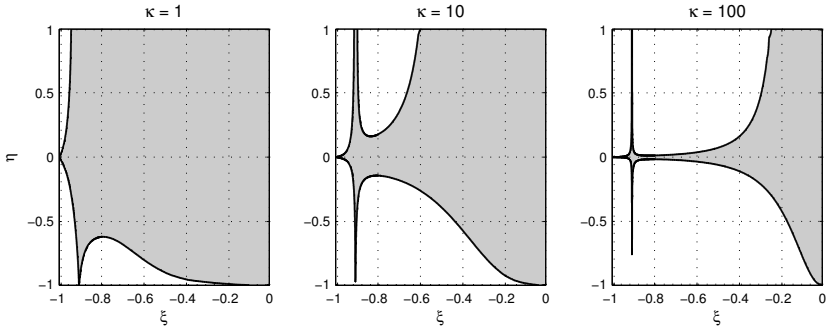


FIGURE 3.7: Recursive refinement, ROS2 with forward quadratic interpolation. Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

step strategy. A curious fact is that for $\kappa = 1$ and $\kappa = 10$ the recursive refinement strategy is stable almost for all the values of η when $\xi = \xi^*$, where ξ^* is a number close to -0.9. For $\kappa = 100$ this property is not valid anymore.

Figure 3.9 shows that the combination of ROS2 and backward quadratic

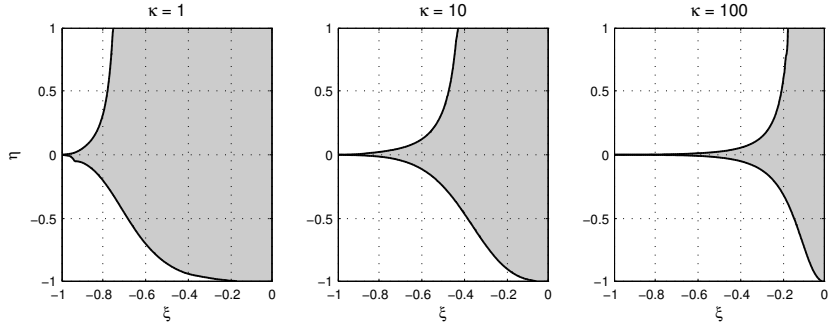


FIGURE 3.8: Compound step, ROS2 with forward quadratic interpolation. Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

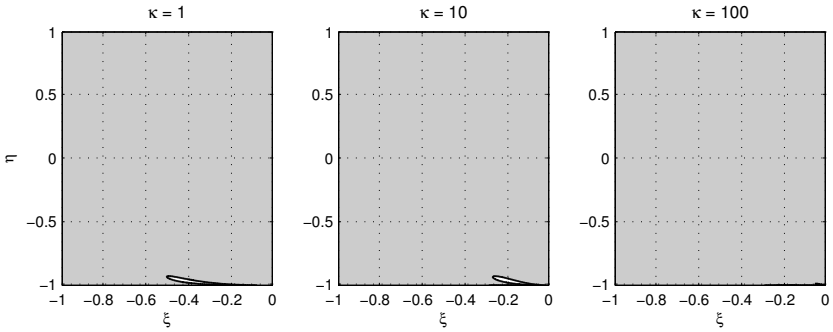


FIGURE 3.9: Recursive refinement, ROS2 with backward quadratic interpolation. Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

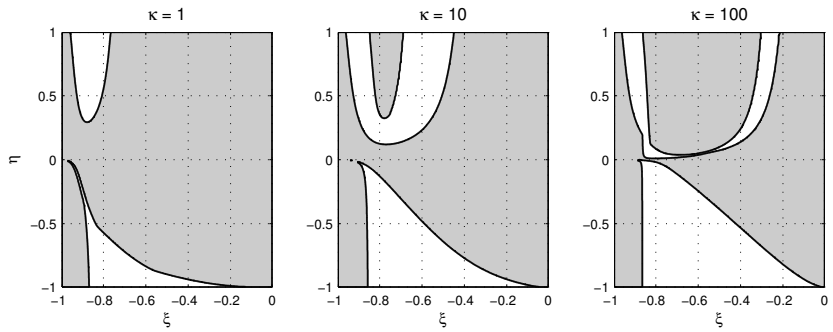


FIGURE 3.10: Compound step, ROS2 with backward quadratic interpolation. Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

interpolation is almost unconditionally stable for the recursive refinement strategy. There is a small set of points in the bottom-right corner of the domain where this strategy is unstable. In this case the stability domain is getting larger with the increase of the stiffness parameter κ, probably due to the L-stability of

the ROS2 scheme. As shown in Figure 3.10, the compound step strategy used with ROS2 and backward quadratic interpolation has large instability regions, which in this case is a disadvantage of this strategy in comparison with the recursive refinement strategy.

In the case of linear and forward quadratic interpolation, for both strategies stability regions decrease with the increase of κ. However, in the case of backward quadratic interpolation, the stability region of the recursive refinement strategy increases with the increase of κ. The compound step strategy, used with backward quadratic interpolation, has irregular large stability regions, which shows that it can lead to unpredictable stability problems.

In this section we showed some results for ROS2 with the choice $\gamma = 1-\frac{1}{2}\sqrt{2}$. We also performed some tests for $\gamma = 1+\frac{1}{2}\sqrt{2}$ and $\gamma = \frac{1}{2}$. The results we obtained are very similar to the ones with $\gamma = 1-\frac{1}{2}\sqrt{2}$. The asymptotic instability regions were a bit larger for $\gamma = 1+\frac{1}{2}\sqrt{2}$ than for $\gamma = 1-\frac{1}{2}\sqrt{2}$. The only significant difference was that ROS2 with $\gamma = \frac{1}{2}$ and backward quadratic interpolation was as unstable as ROS2 with $\gamma = \frac{1}{2}$ and forward quadratic interpolation.

The main result of this section is that for the recursive refinement strategy there exists a second order multirate scheme, based on ROS2 and backward quadratic interpolation, which is unconditionally asymptotically stable (except for a very small region). For the compound step strategy it is not possible to have a second order multirate scheme with this stability property.

3.6 Relevance of the linear 2×2 test problem

Asymptotic stability guarantees $||S^n|| \to 0$ as $n \to \infty$. This also implies boundedness of

$$M = \sup_{n \geq 0} ||S^n|| \,, \tag{3.43}$$

but this bound M may depend on τ and A, and in particular on the stiffness of the problem. There is also lack of theory which would extend the results of stability analysis for multirate schemes for the linear 2×2 test equation to general systems of ODEs. Therefore, in order to see how relevant the asymptotic stability results for the linear 2×2 test problem are we did some stability tests in $\mathbb{R}^m$ to determine M for some interesting matrices A. In this section we consider $m = 50$ and we assume that the first 25 components of the system are fast and the last 25 components are slow. We use ROS2 as our main time integration method. Forward quadratic interpolation showed bad asymptotic stability properties in the 2×2 tests and therefore we do not consider it anymore in the following numerical tests.

3.6.1 The heat equation

Let us consider the heat equation

$$u_t = d u_{xx} \,. \tag{3.44}$$

Applying the second order central discretization on a uniform spatial grid leads to a semi-discrete system

$$w'(t) = Aw(t)\,, \tag{3.45}$$

where A is a $m \times m$ matrix

$$A = \mu \text{ tridiag}(1, -2, 1) \tag{3.46}$$

and $\mu > 0$ will depend on m and d. For matrices A of type (3.46), with $m = 50$, numerical tests for the recursive refinement and compound step strategies based on ROS2 and backward quadratic interpolation showed boundedness for the powers of the amplification matrix of the scheme in the maximum norm. From Figure 3.11 it is seen that in this case $||S^n||_\infty$ is bounded by 2 and 25, for any choice of n and μ, for the recursive refinement and the compound step strategy respectively. The bound value $M = 25$ for the compound step is much larger than $M = 2$ for the recursive refinement strategy. For the compound step strategy M becomes larger with the increase of m; numerical experiments suggest that for this strategy $M = \frac{1}{2}m$, which can be viewed as a weak instability.

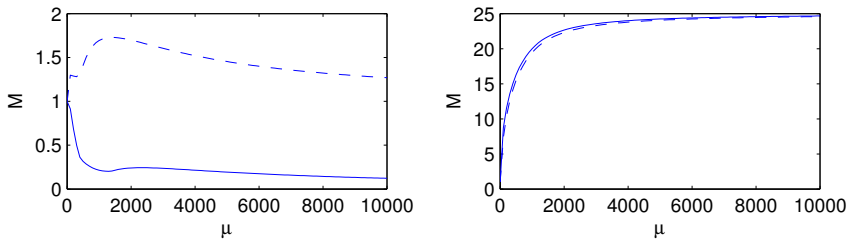


FIGURE 3.11: Problem (3.44). Plot of the bound value M for ROS2 with recursive refinement (left) and compound step (right) strategies, used with linear (solid line) and backward quadratic interpolation (dashed line).

However, if we consider the heat equation with a non-constant diffusion coefficient

$$u_t = d(x)u_{xx} \tag{3.47}$$

then with the same spatial discretization we obtain a semi-discrete system (3.45) with

$$A = \text{diag}(\mu_1, \ldots, \mu_m)\text{tridiag}(1, -2, 1)\,. \tag{3.48}$$

If, for this type of systems, we take $\mu_i = \frac{7}{6}$ for $i \leq 25$ and $\mu_i = \frac{35}{3}$ for $i > 25$ then the compound step strategy based on ROS2 and backward quadratic interpolation becomes unstable. Figure 3.12 shows that for this choice of the coefficients μ_i, $||S^n||_\infty$ is bounded by 2 for any n for the recursive refinement strategy, whereas for the compound step strategy an exponential growth in n is observed.

These numerical results are in accordance with the results obtained for the linear 2×2 test problem. The 2×2 version of the matrix (3.46) would correspond to $\kappa = 1$ and $\eta = \frac{1}{7}$. Figures 3.5, 3.6, 3.9 and 3.10 show that for these values of

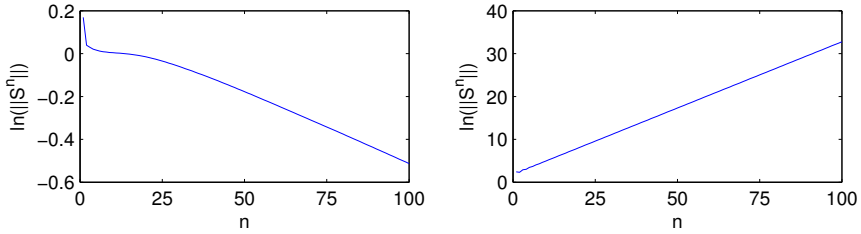


FIGURE 3.12: Problem (3.47). Plot of the $ln(||S^n||)$ for ROS2 with recursive refinement (left) and compound step (right) strategies, used with backward quadratic interpolation.

κ and η both multirate strategies are asymptotically stable. The 2×2 version of the matrix (3.48) corresponds to $\kappa = 10$, $\eta = \frac{1}{7}$ and $\xi = -0.7$. For these values the compound step strategy is asymptotically unstable (Figure 3.10), but the recursive refinement strategy is stable (Figure 3.9).

The numerical tests presented in this subsection suggest that the conclusions obtained in Section 3.5 are also valid for more general systems. The following conjecture can be formulated: the recursive refinement strategy, based on ROS2 and linear or backward quadratic interpolation, is stable if it is applied to the discrete system obtained by second order spatial discretization of the heat equation. In the same context, the compound step strategy is stable if is used with linear interpolation, but it can lead to instabilities when is used with backward quadratic interpolation.

3.6.2 The advection equation

As a second test problem we consider the advection equation

$$u_t + au_x = 0\,. \tag{3.49}$$

Applying the first order upwind discretization on a uniform spatial grid leads to a semi-discrete system

$$w'(t) = Aw(t)\,, \tag{3.50}$$

where A is a $m \times m$ matrix

$$A = \mu\ \text{tridiag}(1, -1, 0)\,. \tag{3.51}$$

For the matrices A of type (3.51), numerical tests for the recursive refinement and compound step strategies based on ROS2 and backward quadratic interpolation showed uniform boundedness for the powers of the amplification matrix of the scheme. From Figure 3.13 it is seen that in this case $||S^n||_\infty$ is bounded by 3 and 35, for any choice of n and μ, for the recursive refinement and the compound step strategy, respectively. The bound $M = 35$ for the compound step strategy is larger than the bound $M = 3$ for the recursive refinement strategy.

However, for this case (3.51) it was observed in further numerical tests that both these bounds do not change significantly, with increasing m, in contrast to (3.46).

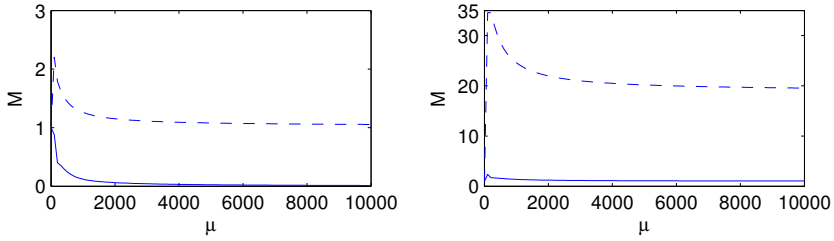


FIGURE 3.13: Problem (3.49), first order upwind spatial discretization. Plot of the bound value M for ROS2 with recursive refinement (left) and compound step (right) strategies, used with linear (solid line) and backward quadratic interpolation (dashed line).

We also consider the case of the second order central spatial discretization of the advection term for the problem (3.49). With this discretization we obtain a semi-discrete system (3.50) with

$$A = \mu \text{ tridiag}(1, 0, -1)\,. \tag{3.52}$$

Numerical tests showed that both multirate strategies used with ROS2 are unstable for the system (3.50) with matrices A of type (3.52). Figure 3.14 shows that the infinity norm of the powers of the amplification matrix S for the case $\mu = 100$ is not bounded.

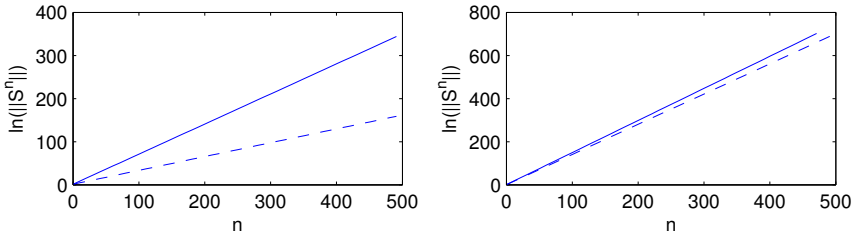


FIGURE 3.14: Problem (3.49), second order central spatial discretization. Plot of the $ln(||S^n||)$ for ROS2 with recursive refinement (left) and compound step (right) strategies, used with linear (solid line) and backward quadratic interpolation (dashed line), ROS2.

Again, the results from this subsection agree with those obtained for the linear 2×2 test problem. The 2×2 version of the matrix (3.51) would correspond to $\kappa = 1$ and $\eta = 0$. Figures 3.5-3.10 show that for these values of κ and η both multirate strategies are asymptotically stable. The 2×2 version of the matrix (3.52) corresponds to $\eta = -1$ and $\xi = 0$. The same Figures show that these

values of κ and ξ can lead to asymptotic instabilities of both strategies. All this suggests that both strategies, based on ROS2 and linear or backward quadratic interpolation, are stable when applied to the semi-discrete system obtained by first order upwind spatial discretization of the advection equation. They are unstable if, instead, the second order central spatial discretization is used.

3.7 A property of the eigenvalues of the amplification matrix for partitioned Rosenbrock methods

All multirate schemes considered in this chapter can be transformed into a partitioned Rosenbrock method, for example by adding some artificial extra stages; see [3], for example.

For a system

$$\begin{aligned} u' &= F_1(u,v)\,, \\ v' &= F_2(u,v)\,, \end{aligned} \tag{3.53}$$

a partitioned Rosenbrock method is given by

$$u_n = u_{n-1} + \sum_{i=1}^{s_1} \bar{b}_i \bar{k}_i\,, \tag{3.54}$$

$$v_n = v_{n-1} + \sum_{i=1}^{s_2} \hat{b}_i \hat{k}_i\,, \tag{3.55}$$

$$\begin{aligned} \bar{k}_i = \tau F_1 &\left(u_{n-1} + \sum_{j=1}^{i-1} \bar{\alpha}_{ij} \bar{k}_j, v_{n-1} + \sum_{j=1}^{\bar{p}_i} \bar{\beta}_{ij} \hat{k}_j \right) \\ &+ \tau F_{1u} \sum_{j=1}^{i} \bar{\gamma}_{ij} \bar{k}_j + \tau F_{1v} \sum_{j=1}^{s_2} \bar{\delta}_{ij} \hat{k}_j, \quad i = 1, \ldots, s_1\,, \end{aligned} \tag{3.56}$$

$$\begin{aligned} \hat{k}_i = \tau F_2 &\left(u_{n-1} + \sum_{j=1}^{\hat{p}_i} \hat{\alpha}_{ij} \bar{k}_j, v_{n-1} + \sum_{j=1}^{i-1} \hat{\beta}_{ij} \hat{k}_j \right) \\ &+ \tau F_{2u} \sum_{j=1}^{s_1} \hat{\gamma}_{ij} \bar{k}_j + \tau F_{2v} \sum_{j=1}^{i} \hat{\delta}_{ij} \hat{k}_j, \quad i = 1, \ldots, s_2\,, \end{aligned} \tag{3.57}$$

where $F_{iu} = \frac{\partial F_i}{\partial u}$ and $F_{iv} = \frac{\partial F_i}{\partial v}$.

We mention that if

$$\bar{p}_i \le i \quad \text{and} \quad \hat{p}_i \le i \tag{3.58}$$

then the system (3.56-3.57) can be solved by sequentially computing the values of the pairs $(\bar{k}_i, \hat{k}_i)$. The recursive refinement strategy leads to a multirate scheme which can be written as a partitioned Rosenbrock method with property (3.58). In the compound step strategy the macro step and the first micro step are computed simultaneously. The micro step uses the information obtained from the interpolation of the results from the macro step. The macro step uses the information obtained by the extrapolation of the results from the micro step. The partitioned Rosenbrock method derived from the multirate scheme obtained with the compound step strategy may not satisfy (3.58). This happens if backward quadratic interpolation is used. In this case all micro steps are computed using interpolation which depends on the value of the solution calculated at the last micro step. Therefore for the compound step strategy, (3.56-3.57) can result in large implicit systems. In practice, backward quadratic interpolation is not used.

In the case of our 2×2 linear test problem the system (3.53) can be written as

$$\begin{aligned} u' &= a_{11}u + a_{12}v\,, \\ v' &= a_{21}u + a_{22}v\,. \end{aligned} \tag{3.59}$$

If we write the method (3.54)-(3.57) in a short form

$$\begin{pmatrix} u_n \\ v_n \end{pmatrix} = S \begin{pmatrix} u_{n-1} \\ v_{n-1} \end{pmatrix}, \tag{3.60}$$

with $S = (S_{ij}), i, j = 1, 2$, then we can prove the following theorem.

Theorem 3.7.1 *The eigenvalues of the amplification matrix S can be written as functions of the three variables z_{11}, z_{22} and* $\det(Z)$.

Proof. For the problem (3.59) the formulas (3.56)-(3.57) reduce to

$$\bar{k}_i = z_{11}(u_{n-1} + \sum_{j=1}^{i} \bar{\alpha}^*_{ij}\bar{k}_j) + z_{12}(v_{n-1} + \sum_{j=1}^{s_2} \bar{\beta}^*_{ij}\hat{k}_j)\,, \tag{3.61}$$

$$\hat{k}_i = z_{21}(u_{n-1} + \sum_{j=1}^{s_1} \hat{\alpha}^*_{ij}\bar{k}_j) + z_{22}(v_{n-1} + \sum_{j=1}^{i} \hat{\beta}^*_{ij}\hat{k}_j)\,. \tag{3.62}$$

If we set $(u_{n-1}, v_{n-1})^T = (1, 0)^T$ then we get $(S_{11}, S_{21})^T = (u_n, v_n)^T$. By defining $\hat{k}_i = z_{21}\hat{k}^*_i$ from (3.61)-(3.62) we obtain

$$\bar{k}_i = z_{11}(1 + \sum_{j=1}^{i} \bar{\alpha}^*_{ij}\bar{k}_j) + z_{12}z_{21}\sum_{j=1}^{s_2} \bar{\beta}^*_{ij}\hat{k}^*_j, \quad i = 1, \ldots, s_1\,, \tag{3.63}$$

$$\hat{k}^*_i = 1 + \sum_{j=1}^{s_1} \hat{\alpha}^*_{ij}\bar{k}_j + z_{22}\sum_{j=1}^{i} \hat{\beta}^*_{ij}\hat{k}^*_j, \quad i = 1, \ldots, s_2\,. \tag{3.64}$$

The solution of system (3.63)-(3.64) depends only on z_{11}, z_{22} and $\det(Z)$. Therefore we have

$$S_{11} = u_n = 1 + \sum_{i=1}^{s_1} \bar{b}_i \bar{k}_i = f_{11}(z_{11}, z_{22}, \det(Z)), \tag{3.65}$$

$$S_{21} = v_n = z_{21} \sum_{i=1}^{s_2} \hat{b}_i \hat{k}_i^* = z_{21} f_{21}(z_{11}, z_{22}, \det(Z)). \tag{3.66}$$

In a similar way, by setting $(u_{n-1}, v_{n-1})^T = (0, 1)^T$ one can show that

$$S_{12} = z_{12} f_{21}(z_{11}, z_{22}, \det(Z)) \text{ and } S_{22} = f_{22}(z_{11}, z_{22}, \det(Z)). \tag{3.67}$$

Finally from

$$S = \begin{pmatrix} f_{11}(z_{11}, z_{22}, \det(Z)) & z_{12} f_{21}(z_{11}, z_{22}, \det(Z)) \\ z_{21} f_{21}(z_{11}, z_{22}, \det(Z)) & f_{22}(z_{11}, z_{22}, \det(Z)) \end{pmatrix} \tag{3.68}$$

the proof of the theorem directly follows. □

This property was already observed for some special methods in [25, 31, 50].

3.8 Conclusions

In this chapter we presented a comparison of asymptotic stability properties for the multirate recursive refinement and the compound step strategies. We also discussed how the obtained results can be used in the context of stability of the more general schemes. For most of the tests in the chapter the recursive refinement strategy does have the asymptotic stability regions somewhat larger than the compound step strategy. Sometimes the difference is very small (ROS1 and quadratic interpolation), in other cases the difference is significant (ROS2 and backward quadratic interpolation).

The scheme based on the recursive refinement strategy used with ROS2 and backward quadratic interpolation is clearly the favorite among the considered second order schemes. It has a very small instability region. There are no multirate schemes based on the compound step strategy, which are of second order for stiff problems and have good stability properties.

The numerical tests for more general systems presented in the chapter gave results which are in accordance with those obtained for the 2×2 linear test problem. Therefore, the simple 2×2 case already gives a good indication for stability properties for more general systems, such as the semi-discrete systems obtained from the spatial discretization of the heat equation and the advection equation.

Finally we mention that the compound step strategy, by avoiding the extra work of doing the macro step for all the components, looses some stability properties compared to the recursive refinement strategy, and it can also lead to more complex implicit systems which are difficult to solve. The recursive refinement strategy is very simple and it has better stability properties.

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.

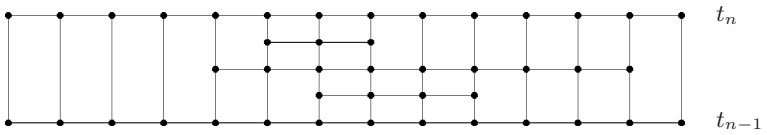


FIGURE 4.1: Multirate time stepping for a time interval $[t_{n-1}, t_n]$.

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 $\gamma \geq \frac{1}{4}, \gamma \neq \frac{1}{2}$.

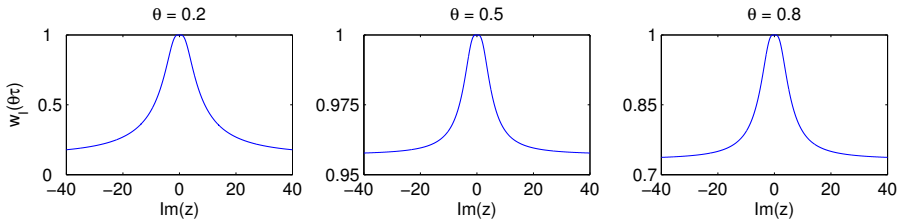


FIGURE 4.2: Plot of $|w_I(\theta\tau)|$ for $\gamma = 1 - \frac{1}{2}\sqrt{2}$ and three different values of θ.

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.

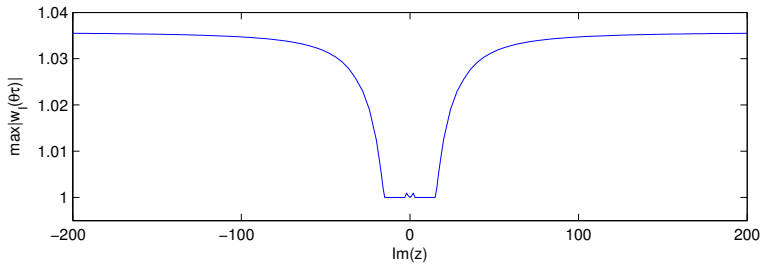


FIGURE 4.3: Plot of the $\max_{0\le\theta\le1}|w_I(\theta\tau)|$ for a range of purely imaginary z-values.

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

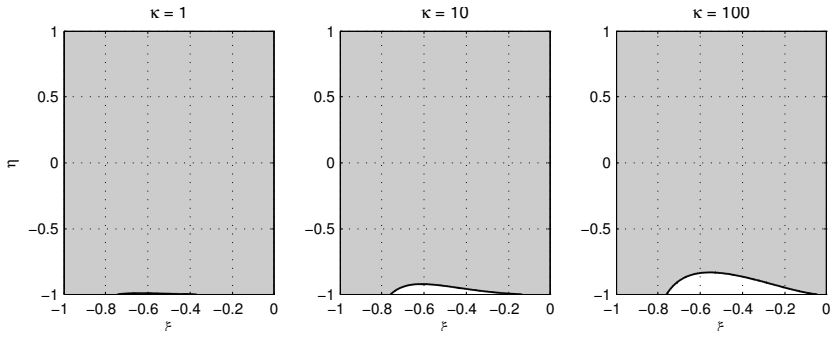


FIGURE 4.4: Asymptotic stability domains (gray areas) for $\kappa = 1, 10, 100$.

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

$$a_2c_2(\theta) + a_3c_3(\theta) + a_4c_4(\theta) = 0\,, \tag{4.27}$$

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.

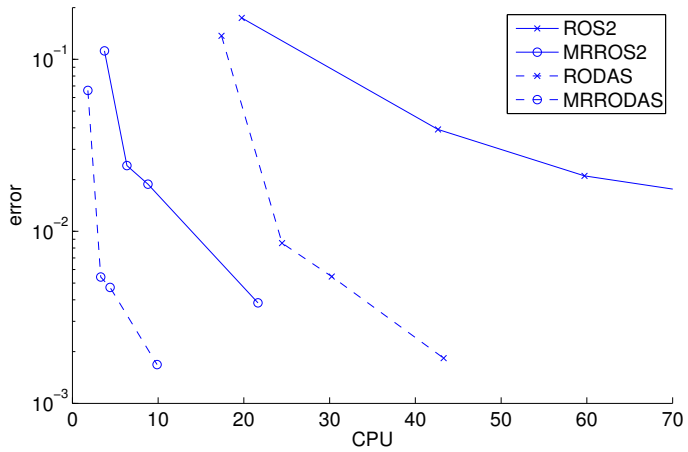


FIGURE 4.5: CPU-error diagram for problem (4.66).

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.

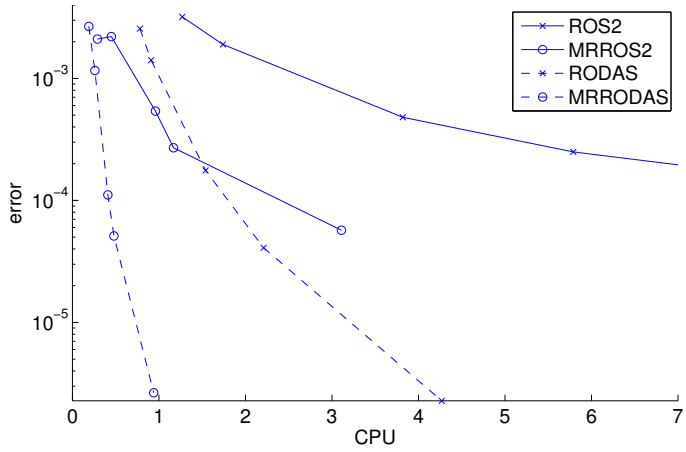


FIGURE 4.6: CPU-error diagram for problem (4.67).

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.

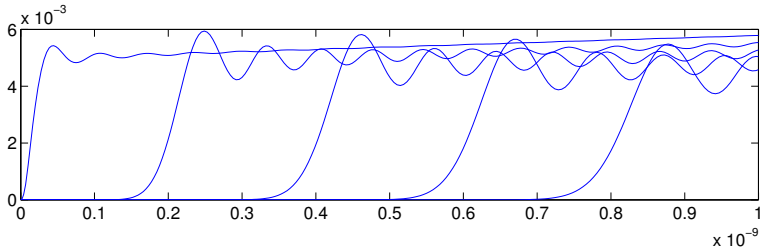


FIGURE 4.7: Solution components v_k, $k = 1, 10, 20, 30, 40$, for problem (4.69).

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.

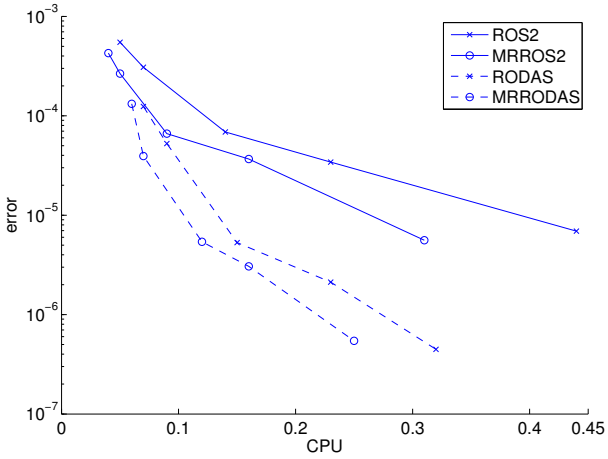


FIGURE 4.8: CPU-error diagram for problem (4.69).

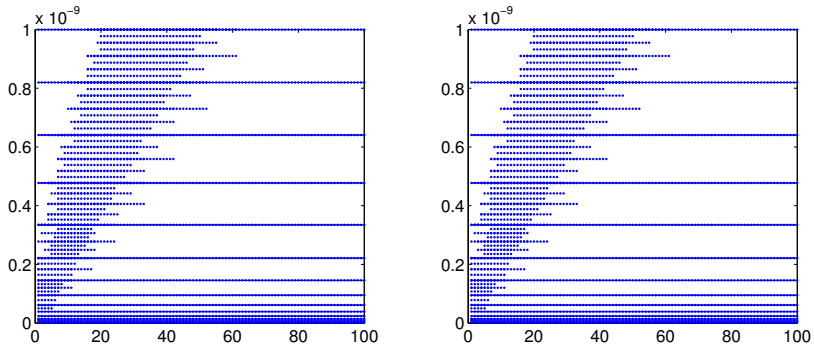


FIGURE 4.9: Component-time grid (v_k left and i_k right) for problem (4.69).

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

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

Chapter 5

Analysis of explicit multirate and partitioned Runge-Kutta schemes for conservation laws

Multirate schemes for conservation laws or convection-dominated problems seem to come in two flavors: schemes that are locally inconsistent, and schemes that lack mass-conservation. In this chapter these two defects are discussed for one-dimensional conservation laws.

Particular attention will be given to monotonicity properties of the multirate schemes, such as maximum principles and the total variation diminishing (TVD) property. The study of these properties will be done within the framework of partitioned Runge-Kutta methods.

5.1 Introduction

Multirate schemes for conservation laws that have appeared in the literature all seem to have one of the following defects: there are schemes that are *locally inconsistent*, e.g. [8, 9, 36, 37], and schemes that are *not mass-conservative*, e.g. [54]. In this chapter these two defects are discussed for one-dimensional conservation laws $u_t + f(u)_x = 0$. We will mainly concentrate on time stepping aspects for simple schemes with one level of temporal refinement. The spatial grids are assumed to be given and fixed in time. Spatial discretization of a PDE (partial differential equation) then leads to a system of ODEs (ordinary differential equations), the so-called semi-discrete system. Particular attention will be given to monotonicity properties of the multirate time stepping schemes, such as maximum principles and the total variation diminishing (TVD) property.

After some preliminaries, we will present in Section 5.3 a detailed analysis of two multirate forward Euler schemes, due to Osher & Sanders [37] and Tang & Warnecke [54]. The first of these schemes is inconsistent at interface points, but it will be shown that convergence of order one can be still obtained in the maximum-norm. Furthermore, we will see that step size restrictions for monotonicity will depend on the type of monotonicity: in general the restrictions for maximum principles can be more relaxed than for the TVD property.

In Section 5.4 we will present some multirate schemes that are based on a standard two-stage Runge-Kutta method. These multirate schemes were recently introduced by Tang & Warnecke [54], Constantinescu & Sandu [8], and Savcenco et al. [47]. For these schemes some results of numerical experiments for linear advection and Burgers' equation are discussed.

For the analysis of general multirate schemes it is convenient to write them in the form of partitioned Runge-Kutta methods. In Section 5.5 it will be seen that recent results for (standard and additive) Runge-Kutta methods of Higueras, Ferracina and Spijker [17, 21, 22, 52] can then be employed to obtain monotonicity results for the multirate schemes through the partitioned Runge-Kutta methods. As for the forward Euler multirate schemes, the step size restrictions for maximum-norm monotonicity and maximum principles are in general more relaxed than for the TVD property. Comparison of the theoretical results with the numerical tests indicates that the restrictions for maximum-norm monotonicity are more relevant in practice. This section also contains a discussion on local and global temporal errors for problems with smooth solutions. To understand the convergence behavior of the schemes, the propagation of the local errors, with associated damping and cancellation effects, are to be taken into account.

5.2 Preliminaries

5.2.1 Forward Euler multirate schemes for the advection equation

Examples of simple schemes

Consider as a simple example the advection equation

$$u_t + u_x = 0 \tag{5.1}$$

on a one-dimensional spatial region $0 < x < 1$ with given initial value $u(x,0)$, and inflow boundary condition $u(0,t)$ or spatial periodicity. Spatial discretization is performed with the first-order upwind scheme on cells $\mathcal{C}_j = (x_j - \frac{1}{2}\Delta x_j, x_j + \frac{1}{2}\Delta x_j)$. This gives a semi-discrete system

$$u_j'(t) = \frac{1}{\Delta x_j}\big(u_{j-1}(t) - u_j(t)\big) \qquad \text{for } j \in \mathcal{I} = \{1, 2, \ldots, m\}\,, \tag{5.2}$$

where $u_j'(t) = \frac{d}{dt}u_j(t)$, and $u_j(t)$ approximates $u(x_j,t)$ or the average value over the surrounding cell $\mathcal{C}_j$.

Application of the forward Euler method with time step Δt gives the CFL stability condition $\nu_j \leq 1$ for all j, where $\nu_j = \Delta t/\Delta x_j$ is the local Courant number. Suppose this stability condition is satisfied for $j \in \mathcal{I}_1$ but on $\mathcal{I}_2 = \mathcal{I} - \mathcal{I}_1$ we need to take two smaller steps with step size $\frac{1}{2}\Delta t$ to reach $t_{n+1} = t_n + \Delta t$.

Then for this simple situation, the scheme of Osher and Sanders [37] can be written as

$$u_j^{n+\frac{1}{2}} = \begin{cases} u_j^n & \text{for } j \in \mathcal{I}_1\,, \\ u_j^n + \frac{1}{2}\nu_j(u_{j-1}^n - u_j^n) & \text{for } j \in \mathcal{I}_2\,, \end{cases} \tag{5.3a}$$

$$u_j^{n+1} = u_j^n + \frac{1}{2}\nu_j(u_{j-1}^n - u_j^n) + \frac{1}{2}\nu_j(u_{j-1}^{n+\frac{1}{2}} - u_j^{n+\frac{1}{2}}) \quad \text{for } j \in \mathcal{I}\,. \tag{5.3b}$$

As observed in [54], the scheme (5.3) is not consistent at the interface: if $i-1 \in \mathcal{I}_1$ and $i \in \mathcal{I}_2$ then

$$\frac{1}{\Delta t}\big(u_i^{n+1} - u_i^n\big) = \frac{1}{\Delta x_i}\big(u_{i-1}^n - \frac{1}{2}(u_i^n + u_i^{n+\frac{1}{2}})\big) = \frac{1 - \frac{1}{4}\nu_i}{\Delta x_i}\big(u_{i-1}^n - u_i^n\big)\,,$$

which is consistent for fixed Courant number ν_i with the equation

$$u_t + (1 - \frac{1}{4}\nu_i)u_x = \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x_i)\,,$$

rather than the original advection equation (5.1).

To overcome this inconsistency, Tang and Warnecke [54] therefore proposed the modified scheme

$$u_j^{n+\frac{1}{2}} = u_j^n + \frac{1}{2}\nu_j(u_{j-1}^n - u_j^n) \quad \text{for } j \in \mathcal{I}\,, \tag{5.4a}$$

$$u_j^{n+1} = u_j^{n+\frac{1}{2}} + \begin{cases} \frac{1}{2}\nu_j(u_{j-1}^n - u_j^n) & \text{for } j \in \mathcal{I}_1\,, \\ \frac{1}{2}\nu_j(u_{j-1}^{n+\frac{1}{2}} - u_j^{n+\frac{1}{2}}) & \text{for } j \in \mathcal{I}_2\,. \end{cases} \tag{5.4b}$$

This scheme, however, is not mass conserving at the interface. If $i-1 \in \mathcal{I}_1$ and $i \in \mathcal{I}_2$ then the flux at $x_{i-1/2}$ that leaves cell $\mathcal{C}_{i-1}$ over the time interval $[t_n, t_{n+1}]$ equals u_{i-1}^n, whereas the flux that enters $\mathcal{C}_i$ is given by $\frac{1}{2}(u_{i-1}^n + u_{i-1}^{n+1/2})$.

It should be noted that except for interface points the schemes (5.3) and (5.4) are identical. For example, if $\mathcal{I}_1 = \{j : j < i\}$ and $\mathcal{I}_2 = \{j : j \geq i\}$, then (5.3) and (5.4) give in one step the same result for $j \neq i$. It will be shown next that, also with larger interface regions, the properties of internal consistency and mass conservation cannot be combined.

Incompatibility of consistency and mass conservation

Consider the first-order upwind discretization (5.2) for the advection equation with spatial periodicity. Then

$$M = \sum_{j \in \mathcal{I}} \Delta x_j u_j(t)\,.$$

is a conserved quantity. If the u_j are densities, this is global mass conservation.

Now suppose that for $j \le k_1$ we use forward Euler with step size Δt, for $j > k_2$ we apply forward Euler with step size $\frac{1}{2}\Delta t$, and on the interface region $k_1 < j \le k_2$ we take any combination of a number of forward Euler steps with Δt and $\frac{1}{2}\Delta t$ together with interpolation or extrapolation. The result can be written as

$$u_j^{n+1} = \begin{cases} u_j^n + \nu_j(u_{j-1}^n - u_j^n)\,, & 1 \le j \le k_1\,, \\ u_j^n + \nu_j(u_{j-1}^n - u_j^n) + \nu_j^2 \sum_{k=1}^{m} \alpha_{jk}\, u_k^n\,, & k_1 < j \le k_2\,, \\ u_j^n + \nu_j(u_{j-1}^n - u_j^n) + \frac{1}{4}\nu_j^2(u_{j-2}^n - 2u_{j-1}^n + u_j^n)\,, & k_2 < j \le m\,, \end{cases} \tag{5.5}$$

with unspecified coefficients α_{jk}, and with $u_0 = u_m$ due to spatial periodicity. The interface at $x = 0, 1$ poses no problem here. We will show that this scheme cannot be both mass conservative and consistent, no matter how the scheme is defined on the interface region $k_1 < j \le k_2$. For convenience it can be assumed that the spatial grid is uniform, $\nu_j = \nu = \Delta t/\Delta x$, and we set $\alpha_{jk} = 0$ for $j \le k_1$ and $j > k_2$.

Insertion of exact solution values in the scheme gives for $k_1 < j \le k_2$ the truncation error

$$\frac{1}{\Delta t}\big(u(x_j, t_{n+1}) - u(x_j, t_n)\big) - \frac{1}{\Delta x}\big(u(x_{j-1}, t_n) - u(x_j, t_n)\big) - \frac{\Delta t}{\Delta x^2}\sum_{k=1}^{m} \alpha_{jk} u(x_k, t_n)\,.$$

For consistency, that is, truncation error $\mathcal{O}(\Delta t) + \mathcal{O}(\Delta x)$, we obtain by Taylor expansion the conditions

$$\sum_k \alpha_{jk} = 0\,, \quad \sum_k (k-j)\alpha_{jk} = 0 \qquad \text{for } k_1 < j \le k_2\,. \tag{5.6}$$

On the other hand, we have

$$\begin{aligned} \Delta x \sum_j u_j^{n+1} - \Delta x \sum_j u_j^n &= \frac{\Delta t^2}{\Delta x} \sum_j \sum_k \alpha_{jk}\, u_k^n + \frac{\Delta t^2}{4\Delta x} \sum_{j > k_2} \big(u_{j-2}^n - 2u_{j-1}^n + u_j^n\big) \\ &= \frac{\Delta t^2}{\Delta x} \sum_k \Big(\sum_j \alpha_{jk}\Big) u_k^n + \frac{\Delta t^2}{4\Delta x} u_{k_2-1}^n - \frac{\Delta t^2}{4\Delta x} u_{k_2}^n\,, \end{aligned}$$

from which it seen that the requirement of mass conservation leads to

$$\sum_j \alpha_{jk} = \begin{cases} 0 & \text{if } k \ne k_2 - 1, k_2\,, \\ -\frac{1}{4} & \text{if } k = k_2 - 1\,, \\ \frac{1}{4} & \text{if } k = k_2\,. \end{cases} \tag{5.7}$$

However, the conditions (5.6) and (5.7) together lead to a contradiction:

$$0 \;=\; \sum_j \sum_k (k-j)\alpha_{jk} \;=\; \sum_j \sum_k \big((k-k_2+1)-(j-k_2+1)\big)\alpha_{jk}$$
$$= \sum_j (j-k_2+1) \sum_k \alpha_{jk} - \sum_k (k-k_2+1) \sum_j \alpha_{jk} \;=\; \sum_j \alpha_{jk_2} \;=\; \frac{1}{4}\,.$$

This shows that consistency and mass conservation cannot be valid at the same time.

5.2.2 General formulations

In this chapter we will discuss monotonicity properties and temporal convergence of multirate schemes for general semi-discrete problems in $\mathbb{R}^m$,

$$u'(t) = F(u(t))\,, \qquad u(0) = u_0\,. \tag{5.8}$$

The approximations to $u(t_n) = [u_j(t_n)] \in \mathbb{R}^m$ will be denoted by $u_n = [u_j^n] \in \mathbb{R}^m$. As above, we consider partitioning $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$. Corresponding to these sets $\mathcal{I}_k$, let I_1, I_2 be $m \times m$ diagonal matrices with diagonal entries 0 or 1, such that $(I_k)_{jj} = 1$ for $j \in \mathcal{I}_k$, $k = 1, 2$. We have $I_1 + I_2 = I$, the identity matrix.

The semi-discrete system (5.2) obviously fits in this form with linear F. The general system (5.8) allows nonlinear problems and nonlinear discretizations. For such systems the Osher-Sanders scheme (5.3) becomes

$$\begin{cases} u_{n+\frac{1}{2}} = u_n + \frac{1}{2}\Delta t I_2 F(u_n)\,, \\ u_{n+1} = u_n + \frac{1}{2}\Delta t F(u_n) + \frac{1}{2}\Delta t F(u_{n+\frac{1}{2}})\,, \end{cases} \tag{5.9}$$

and the Tang-Warnecke scheme (5.4) reads

$$\begin{cases} u_{n+\frac{1}{2}} = u_n + \frac{1}{2}\Delta t F(u_n)\,, \\ u_{n+1} = u_n + \Delta t I_1 F(u_n) + \frac{1}{2}\Delta t I_2\big(F(u_n) + F(u_{n+\frac{1}{2}})\big)\,. \end{cases} \tag{5.10}$$

In the following we will refer to (5.9) as the OS1 scheme, and to (5.10) as the TW1 scheme. We note that in [37] and [54] the number of sub-steps on the index set $\mathcal{I}_2$ was allowed to be larger than two for these schemes. More general formulations will be considered in Section 5.5.

5.2.3 Monotonicity assumptions

Consider a suitable convex functional,[1] semi-norm or norm $\|v\|$ for $v = [v_j] \in \mathbb{R}^m$. Interesting examples are the maximum-norm

$$\|v\|_\infty = \max_{1\le j\le m} |v_j|\,, \tag{5.11}$$

[1] Recall that $\phi : \mathbb{R}^m \to \mathbb{R}$ is called a convex functional if $\phi(v) \ge 0$, $\phi(v+w) \le \phi(v) + \phi(w)$ and $\phi(\alpha v) = \alpha\phi(v)$ for all $\alpha \ge 0$, $v, w \in \mathbb{R}^m$. If we also have $\phi(-v) = \phi(v)$ for all $v \in \mathbb{R}^m$, then ϕ is a semi-norm. If it holds in addition that $\phi(v) = 0$ only if $v = 0$, then ϕ is a norm.

or the total variation semi-norm

$$\|v\|_{\text{TV}} = \sum_{j=1}^{m} |v_{j-1} - v_j| \qquad \text{with } v_0 = v_m \,, \tag{5.12}$$

arising from one-dimensional scalar PDEs with spatial periodicity.

The basic monotonicity assumption on the semi-discrete system that will be used in this section is

$$\|v + \tau_1 I_1 F(v) + \frac{1}{2}\tau_2 I_2 F(v)\| \le \|v\| \qquad \text{for all } v \in \mathbb{R}^m \text{ and } 0 \le \tau_1, \tau_2 \le \tau_0 \,, \tag{5.13}$$

where $\tau_0 > 0$ is a problem dependent parameter. For the multirate schemes we shall determine factors C such that we have the monotonicity property

$$\|u_{n+1}\| \le \|u_n\| \quad \text{whenever } \Delta t \le C\tau_0 \,. \tag{5.14}$$

For a given scheme, the optimal C will be called the threshold factor for monotonicity. In general, such monotonicity properties are intended to ensure that unwanted overshoots or numerical oscillations will not arise. Following [48, 49] we will call a scheme total variation diminishing (TVD) if (5.14) holds with the semi-norm (5.12). If the (semi-)norm is not specified, methods that have a positive threshold C can be called strong stability preserving (SSP), as in [15] for standard, single-rate methods.

Example 5.2.1 Apart from (semi-)norms, such as $\|v\|_{\text{TV}}$ and $\|v\|_\infty$, we can also consider sublinear functionals. For example, following [52], consider

$$\|v\|_+ = \max_{1\le j\le m} v_j \,, \qquad \|v\|_- = -\min_{1\le j\le m} v_j \,.$$

Then, having (5.14) for both these convex functionals amounts to the maximum principle

$$\min_{1\le i\le m} u_i^0 \;\le\; u_j^n \;\le\; \max_{1\le i\le m} u_i^0 \qquad \text{for all } n \ge 1 \text{ and } 1 \le j \le m \,.$$

In general, this is of course somewhat stronger than having monotonicity in the maximum-norm, $\|u_{n+1}\|_\infty \le \|u_n\|_\infty$, but for the schemes considered in this chapter the associated threshold values C will be the same. □

Example 5.2.2 Consider a scalar conservation law $u_t + f(u)_x = 0$ with a periodic boundary condition, and with $0 \le f'(u) \le \alpha$. Spatial discretization in conservation form gives semi-discrete systems (5.8) with

$$F_j(v) = \frac{1}{\Delta x_j}\big(f(v_{j-\frac{1}{2}}) - f(v_{j+\frac{1}{2}})\big)$$

where $v_{j\pm 1/2}$ are the values at the cell boundaries, determined from the components of $v = [v_i] \in \mathbb{R}^m$. Using limiters in the discretization it can be guaranteed that

$$0 \;\le\; \frac{v_{j-\frac{1}{2}} - v_{j+\frac{1}{2}}}{v_{j-1} - v_j} \;\le\; 1 + \mu$$

with a constant $\mu \geq 0$ determined by the limiter; see also formula (8) in [9]. This holds trivially for the first-order upwind discretization with $\mu = 0$; a detailed higher-order example will be given in Appendix. It now follows that $F_j(v)$ can be written as

$$F_j(v) = \frac{a_j(v)}{\Delta x_j}\big(v_{j-1} - v_j\big)\,, \qquad j = 1, \ldots, m\,, \;\; v_0 = v_m\,,$$

where

$$0 \leq a_j(v) \leq \alpha(1+\mu) \qquad \text{for all } j \text{ and } v \in \mathbb{R}^m.$$

Suppose that $\Delta x_j = h$ for $j \in \mathcal{I}_1$ and $\Delta x_j = \frac{1}{2}h$ for $j \in \mathcal{I}_2$. Then a well-known lemma of Harten [20, Lemma 2.2] shows that (5.13) will be valid for the total variation semi-norm (5.12) provided that

$$\frac{\alpha\tau_0}{h} \;\leq\; \frac{1}{1+\mu}\,.$$

Moreover, it is easy to see that (5.13) will also hold in the maximum-norm under the same CFL restriction. □

5.3 Analysis of the forward Euler multirate schemes

5.3.1 Monotonicity results

Monotonicity results for scheme TW1

Standard (single-rate) schemes give the same step size restriction for various monotonicity properties. As we shall see, with the multirate schemes different step size restrictions are obtained for the maximum-norm or the total variation semi-norm.

In the first stage of the TW1 scheme (5.10) we have of course

$$\|u_{n+\frac{1}{2}}\| \leq \|u_n\| \qquad \text{whenever } \Delta t \leq \tau_0\,.$$

The second stage can be written in the form

$$u_{n+1} = (1-\theta)u_n + \theta\big(u_{n+\frac{1}{2}} - \frac{1}{2}\Delta t F(u_n)\big) + \Delta t I_1 F(u_n) + \frac{1}{2}\Delta t I_2\big(F(u_n) + F(u_{n+\frac{1}{2}})\big)\,,$$

with arbitrary $\theta \in [0,1]$. This leads to

$$\begin{aligned} u_{n+1} \;=\; & (1-\theta)\Big(u_n + \frac{2-\theta}{2(1-\theta)}\Delta t I_1 F(u_n) + \frac{1}{2}\Delta t I_2 F(u_n)\Big) \\ & + \theta\Big(u_{n+\frac{1}{2}} + \frac{1}{2\theta}\Delta t I_2 F(u_{n+\frac{1}{2}})\Big)\,. \end{aligned} \tag{5.15}$$

Under assumption (5.13) this gives the monotonicity property (5.14) with

$$C \;=\; \max_{0\leq\theta\leq 1} \min\Big(1\,,\; \frac{2(1-\theta)}{2-\theta}\,,\, \theta\Big) \;=\; 2 - \sqrt{2}\,. \tag{5.16}$$

This value $C \approx 0.58$ is valid for general semi-norms. So, in particular, it provides a TVD result for schemes with limiters.

Next, consider the maximum-norm. Then, by noting that the second stage can also be written as

$$u_{n+1} = I_1\big(u_n + \Delta t I_1 F(u_n)\big) + I_2\big(u_{n+\frac{1}{2}} + \frac{1}{2}\Delta t I_2 F(u_{n+\frac{1}{2}})\big)\,,$$

it directly follows (see also [54, Lemma 2.1]) that the threshold factor for max-norm monotonicity is

$$C = 1\,. \tag{5.17}$$

Note that this result has been obtained by using the inequality

$$\|I_1 v + I_2 w\| \le \max(\|v\|, \|w\|)\,, \tag{5.18}$$

which holds for the maximum-norm and for the convex functionals $\|\cdot\|_\pm$ from Example 5.2.1, but not for general norms or semi-norms; in particular, it will not hold for the total variation semi-norm.

Monotonicity results for scheme OS1

In the first stage of the OS1 scheme (5.9) we directly obtain

$$\|u_{n+\frac{1}{2}}\| \le \|u_n\| \qquad \text{whenever } \Delta t \le \tau_0\,.$$

The second stage can be written as

$$u_{n+1} = (1-\theta)u_n + \theta\big(u_{n+\frac{1}{2}} - \frac{1}{2}\Delta t I_2 F(u_n)\big) + \frac{1}{2}\Delta t F(u_n) + \frac{1}{2}\Delta t F(u_{n+\frac{1}{2}})$$

with parameter $\theta \in [0,1]$. Hence

$$\begin{aligned} u_{n+1} \;=\; & (1-\theta)\Big(u_n + \tfrac{1}{2(1-\theta)}\Delta t I_1 F(u_n) + \tfrac{1}{2}\Delta t I_2 F(u_n)\Big) \\ & + \theta\Big(u_{n+\frac{1}{2}} + \tfrac{1}{2\theta}\Delta t F(u_{n+\frac{1}{2}})\Big)\,. \end{aligned} \tag{5.19}$$

It follows that under assumption (5.13) the monotonicity property (5.14) holds with

$$C \;=\; \max_{0\le\theta\le 1} \min\big(1\,,\,2(1-\theta)\,,\,\theta\big) \;=\; \frac{2}{3}\,. \tag{5.20}$$

Again, for the maximum-norm a better result can be obtained by considering $I_1 u_{n+1}$ and $I_2 u_{n+1}$ separately. Multiplication of (5.19) with I_1 and taking $\theta = \theta_1 = \frac{1}{2}$ gives

$$I_1 u_{n+1} = \frac{1}{2} I_1\big(u_n + \Delta t I_1 F(u_n)\big) + \frac{1}{2} I_1\big(u_{n+\frac{1}{2}} + \Delta t I_1 F(u_{n+\frac{1}{2}})\big)\,.$$

Likewise, with $\theta = \theta_2 = 1$, it follows that

$$I_2 u_{n+1} = I_2\big(u_{n+\frac{1}{2}} + \frac{1}{2}\Delta t I_2 F(u_{n+\frac{1}{2}})\big)\,.$$

Hence the threshold factor for max-norm monotonicity is

$$C = 1\,. \tag{5.21}$$

This result, formulated in terms of a maximum principle, was already obtained in [37] for first-order upwind spatial discretization and in [30] for a class of high-resolution discretizations. In these papers also TVD results were presented; this will be discussed below.

The TVD property for linear first-order upwind advection

For the linear advection equation $u_t + u_x = 0$ with spatial periodicity, the first-order upwind discretization (5.2) can be written as

$$u'(t) = Au(t)\,, \qquad A = H^{-1}(E - I)\,, \tag{5.22}$$

with $H = \mathrm{diag}(\Delta x_1, \ldots, \Delta x_m)$ and E the backward shift operator, $(Ev)_i = v_{i-1}$ for $i = 1, \ldots, m$ with $v_0 = v_m$. Consider also

$$\tilde{A} = H^{-1}(-I + E^T)\,.$$

This corresponds to first-order upwind discretization for $u_t - u_x = 0$. We denote $Z = \Delta t A$, $\tilde{Z} = \Delta t \tilde{A}$. Then

$$\tilde{Z} = H^{-1} Z^T H\,.$$

For the OS1 and TW1 schemes applied to (5.22) we have $u_{n+1} = S u_n$, where the amplification matrix S can be written as $S = R(Z)$ with

$$R(Z) = \begin{cases} R_{\text{OS1}}(Z) = I + Z + \frac{1}{4} Z I_2 Z\,, \\ R_{\text{TW1}}(Z) = I + Z + \frac{1}{4} I_2 Z^2\,. \end{cases}$$

Let $\tilde{R}$ be such that

$$\tilde{R}(Z)\, Z = Z\, R(Z)\,. \tag{5.23}$$

It is easily seen that $\tilde{R}_{\text{OS1}}(Z) = I + Z + \frac{1}{4} Z^2 I_2$ and $\tilde{R}_{\text{TW1}}(Z) = I + Z + \frac{1}{4} Z I_2 Z$. For both schemes it follows by some simple calculations that

$$R(\tilde{Z}) = H^{-1} \tilde{R}(Z)^T H\,. \tag{5.24}$$

As we saw above, both schemes OS1 and TW1 are such that

$$\|R(\tilde{Z})\|_\infty \le 1 \tag{5.25}$$

whenever $\nu_j = \Delta t / \Delta x_j \le k$ for $j = \mathcal{I}_k$, $k = 1, 2$. It will now be demonstrated that under the same CFL restriction we have

$$\|R(Z)v\|_{\text{TV}} \le \|v\|_{\text{TV}} \qquad \text{for all } v \in \mathbb{R}^m, \tag{5.26}$$

that is, the TVD property is valid with threshold $C = 1$ for the special case of first-order upwind advection discretization.

Lemma 5.3.1 *If (5.24) and (5.25) are valid, then (5.26) is also satisfied.*

Proof. Along with the discrete L_1-norm on $\mathbb{R}^m$, $\|v\|_1 = \sum_{j=1}^m \Delta x_j |v_j|$, we also consider the ℓ_1-norm $\|v\|_{\ell_1} = \sum_{j=1}^m |v_j|$, together with the induced matrix norms. Then we have $\|W\|_\infty = \|W^T\|_{\ell_1}$ for any $W \in \mathbb{R}^{m\times m}$; see for example [23]. Moreover, it is easily seen that $\|W^T\|_{\ell_1} = \|H^{-1}W^T H\|_1$, and therefore

$$\|W\|_\infty = \|H^{-1}W^T H\|_1 \,.$$

Hence (5.24) and (5.25) imply

$$\|\tilde{R}(Z)\|_1 \le 1 \,. \tag{5.27}$$

Further we have

$$\|v\|_{\text{TV}} = \sum_{j=1}^m |v_{j-1} - v_j| = \|Av\|_1 = \frac{1}{\Delta t}\|Zv\|_1 \,.$$

Consequently, for a scheme $u_{n+1} = R(Z)u_n$ the TVD property (5.26) is equivalent to

$$\|ZR(Z)v\|_1 = \|\tilde{R}(Z)Zv\|_1 \le \|Zv\|_1 \,.$$

This is satisfied because $\|\tilde{R}(Z)w\|_1 \le \|w\|_1$ for any $w \in \mathbb{R}^m$, in view of (5.27). □

The above result is not new for the OS1 scheme. In fact, already in [37] the result was given for the case of first-order upwind discretization for nonlinear problems. In [30] this was extended to a class of high-resolution spatial discretizations. The proofs of these more general results for the OS1 scheme are more technical than the above.

5.3.2 Convergence for smooth problems

In this section bounds for the global errors $e_n = u(t_n) - u_n$ will be derived. It will be assumed that the problem (5.8) is sufficiently smooth. Both the schemes OS1 and TW1 are covered by the formula

$$\begin{aligned} u_{n+\frac12} &= u_n + \kappa\Delta t I_1 F(u_n) + \tfrac12 \Delta t I_2 F(u_n) \,, \\ u_{n+1} &= u_n + \tfrac12 \Delta t \big(F(u_n) + F(u_{n+\frac12})\big) + \kappa\Delta t I_1 \big(F(u_n) - F(u_{n+\frac12})\big) \,, \end{aligned} \tag{5.28}$$

with parameter value $\kappa = 0$ for OS1 and $\kappa = \frac12$ for TW1.

If we insert exact ODE values $u(t_n)$, $u(t_{n+1/2})$, $u(t_{n+1})$ into the stages of (5.28) we obtain residuals $\rho_{n+1/2}$ and ρ_{n+1}, respectively. By Taylor expansions it is easily found that

$$\begin{aligned} \rho_{n+\frac12} &= u(t_{n+\frac12}) - u(t_n) - \kappa\Delta t I_1 u'(t_n) - \tfrac12 \Delta t I_2 u'(t_n) \\ &= \big(\tfrac12 - \kappa\big)\Delta t I_1 u'(t_n) + \tfrac18 \Delta t^2 u''(t_n) + \mathcal{O}(\Delta t^3) \,, \end{aligned}$$

$$\rho_{n+1} = u(t_{n+1}) - u(t_n) - \big(\tfrac{1}{2}I + \kappa I_1\big)\Delta t u'(t_n) - \big(\tfrac{1}{2}I - \kappa I_1\big)\Delta t u'(t_{n+\frac{1}{2}})$$
$$= \Delta t^2\big(\tfrac{1}{4}I + \tfrac{1}{2}\kappa I_1\big)u''(t_n) + \mathcal{O}(\Delta t^3)\,.$$

Let $Z_\ell \in \mathbb{R}^{m\times m}$ be such that

$$Z_\ell\big(u(t_\ell) - u_\ell\big) = \Delta t\big(F(u(t_\ell)) - F(u_\ell)\big) \tag{5.29}$$

for all $\ell = n, n+\frac{1}{2}$, $n \geq 0$. If F is differentiable we can take Z_ℓ as the integrated Jacobian matrix

$$Z_\ell = \int_0^1 \Delta t F'(\theta u(t_\ell) + (1-\theta)u_\ell)\,d\theta\,.$$

For the errors in the two stages of (5.28) it follows that

$$e_{n+\frac{1}{2}} = e_n + \kappa I_1 Z_n e_n + \tfrac{1}{2}I_2 Z_n e_n + \rho_{n+\frac{1}{2}}\,,$$
$$e_{n+1} = e_n + \tfrac{1}{2}Z_n e_n + \tfrac{1}{2}Z_{n+\frac{1}{2}}e_{n+\frac{1}{2}} + \kappa I_1\big(Z_n e_n - Z_{n+\frac{1}{2}}e_{n+\frac{1}{2}}\big) + \rho_{n+1}\,.$$

Eliminating $e_{n+1/2}$ we thus obtain a recursion for the global errors of the form

$$e_{n+1} = S_n e_n + d_n\,, \qquad n = 0, 1, \ldots\,, \tag{5.30}$$

with amplification matrix S_n and local discretization error d_n. The resulting expressions are given below for $\kappa = 0, \frac{1}{2}$. The recursion (5.30) will be the basis for the subsequent analysis. The method is called *consistent* of order p if $\|d_n\| = \mathcal{O}(\Delta t^{p+1})$, and *convergent* of order p if $\|e_n\| = \mathcal{O}(\Delta t^p)$ for all n.

Since we want to study convergence at all grid points, including the interface points, the natural norm is the maximum-norm. For stability it will be assumed that

$$\|I + I_1 Z_\ell + \frac{1}{2}I_2 Z_\ell\|_\infty \leq 1\,, \tag{5.31}$$

for all $\ell = n, n+\frac{1}{2}$. It is easily seen that we then have

$$\|I + \theta_1 I_1 Z_\ell + \frac{1}{2}\theta_2 I_2 Z_\ell\|_\infty \leq 1$$

whenever $0 \leq \theta_j \leq 1$. This is of the same form as (5.13), with $F(v)$ replaced by $Z_\ell v$.

In combination with the smoothness assumptions on the problem this stability result will easily lead to convergence for the TW1 scheme. Due to the inconsistency at interface points, the error build-up is more complicated for scheme OS1. It will still be possible to show convergence with order one under the following additional assumptions:

$$\|I_2 Z_\ell\|_\infty \leq 4K < 4\,, \tag{5.32}$$

$$\|Z_{\ell+\frac{1}{2}} - Z_\ell\|_\infty \leq L\Delta t\,, \tag{5.33}$$

for $\ell = n, n+\frac{1}{2}$, $n \geq 0$, with constants $K \in (0,1)$ and $L \geq 0$. Note that (5.32) may be slightly stronger than the local CFL condition implied by (5.31) on the index set $\mathcal{I}_2$.

Convergence of scheme TW1

For the TW1 scheme (5.10) we obtain from the above derivation, with $\kappa = \frac{1}{2}$, the expressions

$$S_n = I_1\big(I + Z_n\big) + I_2\big(I + \frac{1}{2}Z_{n+\frac{1}{2}}\big)\big(I + \frac{1}{2}Z_n\big)\,, \tag{5.34}$$

$$d_n = \frac{1}{2}\Delta t^2\big(I_1 + \frac{1}{2}I_2 + \frac{1}{8}I_2 Z_{n+\frac{1}{2}}\big)u''(t_n) + \mathcal{O}(\Delta t^3)\,. \tag{5.35}$$

As already noted above, (5.31) has the same form as (5.13). Therefore we can copy the derivation leading to (5.17) which now gives the bound

$$\|S_n\|_\infty \le 1 \tag{5.36}$$

for the amplification matrix.

Furthermore, (5.31) implies $\|I_2(I + \frac{1}{4}Z_\ell)\|_\infty \le 1$, which provides the local error bound

$$\|d_n\|_\infty \le \frac{1}{2}\Delta t^2\|u''(t_n)\|_\infty + \mathcal{O}(\Delta t^3)\,.$$

Convergence now follows in a standard fashion. Summarizing, we have the following result:

Theorem 5.3.1 *Consider the TW1 scheme (5.10) with the time step restriction (5.31). Then $\|S\|_\infty \le 1$, and we have the error bound*

$$\|e_n\|_\infty \;\le\; \frac{1}{2}T\Delta t \max_{t\in[0,T]}\|u''(t)\|_\infty + \mathcal{O}(\Delta t^2)\,, \qquad 0 \le t_n \le T\,.$$

Convergence of scheme OS1

Also for the OS1 scheme (5.9) we can prove convergence with order one in the maximum-norm, in spite of the local inconsistencies. For this result, damping and cancellation effects are to be taken into account.

For the OS1 scheme we obtain from the above derivation, with $\kappa = 0$, the expressions

$$S_n = I + \frac{1}{2}Z_n + \frac{1}{2}Z_{n+\frac{1}{2}}\big(I + \frac{1}{2}I_2 Z_n\big)\,, \tag{5.37}$$

$$d_n = \frac{1}{4}\Delta t Z_{n+\frac{1}{2}}I_1 u'(t_n) + \frac{1}{4}\Delta t^2\big(I + \frac{1}{4}Z_{n+\frac{1}{2}}\big)u''(t_n) + \mathcal{O}(\Delta t^3)\,. \tag{5.38}$$

In the same way as above it follows that (5.36) is valid, showing stability of the error recursion. However, here we get only an $\mathcal{O}(\Delta t)$ bound for the local errors because $Z_\ell I_1 u'(t_n)$ will not be an $\mathcal{O}(\Delta t)$ term in general; this is due to the fact that $I_1 u'(t)$ is not a smooth grid function (jumps at the interfaces). To prove convergence we need to establish a relation between local errors and amplification factors.

We have

$$S_n - I = Z_{n+\frac{1}{2}}\big(I + \frac{1}{4}I_2 Z_n\big) - \frac{1}{2}\big(Z_{n+\frac{1}{2}} - Z_n\big)\,.$$

Hence

$$Z_{n+\frac{1}{2}} = (S_n - I)Q_n + \tfrac{1}{2}\big(Z_{n+\frac{1}{2}} - Z_n\big)Q_n\,, \qquad Q_n = \big(I + \tfrac{1}{4}I_2 Z_n\big)^{-1}\,.$$

It follows that we can decompose the local error as

$$d_n = (S_n - I)\xi_n + \eta_n\,, \tag{5.39}$$

with

$$\xi_n = \tfrac{1}{4}\Delta t Q_n I_1 u'(t_n)\,,$$

$$\eta_n = \tfrac{1}{8}\Delta t\big(Z_{n+\frac{1}{2}} - Z_n\big)Q_n I_1 u'(t_n) + \tfrac{1}{4}\Delta t^2\big(I + \tfrac{1}{4}Z_{n+\frac{1}{2}}\big)u''(t_n) + \mathcal{O}(\Delta t^3)\,. \tag{5.40}$$

Such a decomposition can be used to show convergence for scheme OS1; the arguments are the same as in [27, p. 216] for constant $S_n = S$. Let us define $\hat{e}_n = e_n + \xi_n$ for $n \geq 0$. Then

$$\hat{e}_{n+1} = S_n \hat{e}_n + \hat{d}_n\,, \qquad \hat{d}_n = \xi_{n+1} - \xi_n + \eta_n\,,$$

for $n \geq 0$. Hence

$$\|\hat{e}_n\|_\infty \leq \|\hat{e}_0\|_\infty + \sum_{k=0}^{n} \|\hat{d}_k\|_\infty\,.$$

Since $e_0 = 0$ we obtain

$$\|e_n\|_\infty \leq \|\xi_0\|_\infty + \|\xi_n\|_\infty + \sum_{k=0}^{n} \big(\|\xi_{k+1} - \xi_k\|_\infty + \|\eta_k\|_\infty\big)\,. \tag{5.41}$$

It remains to bound the terms on the right-hand side. Under assumption (5.32) it is easily seen that

$$\|Q_k\|_\infty \leq (1 - K)^{-1}\,.$$

Moreover, we have

$$Q_{k+1} - Q_k = -\frac{1}{4}Q_k(I_2 Z_{k+1} - I_2 Z_k)Q_{k+1}\,,$$

$$\|Q_{k+1} - Q_k\|_\infty \leq \frac{1}{2}\Delta t L (1 - K)^{-2}\,.$$

It follows that

$$\|\xi_k\|_\infty \leq \tfrac{1}{4}(1 - K)^{-1}\Delta t \|u'(t_k)\|_\infty\,,$$

$$\|\xi_{k+1} - \xi_k\|_\infty \leq \tfrac{1}{8}(1 - K)^{-2} L \Delta t^2 \|u'(t_k)\|_\infty + \tfrac{1}{4}(1 - K)^{-1}\Delta t^2 \|u''(t_k)\|_\infty + \mathcal{O}(\Delta t^3)\,,$$

$$\|\eta_k\|_\infty \leq \tfrac{1}{8}(1 - K)^{-1} L \Delta t^2 \|u'(t_k)\|_\infty + \tfrac{1}{4}\Delta t^2 \|u''(t_k)\|_\infty\,.$$

Insertion of these three estimates into (5.41) gives the following convergence result.

Theorem 5.3.2 *Consider the OS1 scheme (5.9) with the time step restriction (5.31). Then* $\|S\|_\infty \le 1$. *Under the additional assumption (5.32), (5.33) we have the error bound*

$$\|e_n\|_\infty \;\le\; (M_1 + M_2 TL)\Delta t \max_{t\in[0,T]} \|u'(t)\|_\infty + M_3 T\Delta t \max_{t\in[0,T]} \|u''(t)\|_\infty + \mathcal{O}(\Delta t^2)\,,$$

for $0 \le t_n \le T$, *with* M_1, M_2, M_3 *determined by* K.

Convergence of OS1 for linear first-order upwind advection

Consider the first-order upwind discretization (5.2) for linear advection. Then (5.31) will hold if

$$\frac{\Delta t}{\Delta x_j} \le 1 \quad \text{for } j \in \mathcal{I}_1\,, \qquad \frac{\Delta t}{2\Delta x_j} \le 1 \quad \text{for } j \in \mathcal{I}_2\,.$$

These are the usual restrictions on the local Courant numbers. To have (5.32) we get the restriction

$$\frac{\Delta t}{2\Delta x_j} \le K < 1 \;\text{ for } j \in \mathcal{I}_2\,.$$

However, for this first-order upwind advection case the condition (5.32) with $K < 1$ is not needed. Let $Z = \Delta t A$ with A as in (5.22). Suppose for simplicity that $\mathcal{I}_1 = \{j : j < i\}$, $\mathcal{I}_2 = \{j : j \ge i\}$ with given $i \in \mathcal{I}$. Consider

$$(S - I)\xi = ZI_1 v\,,$$

where $\xi = \xi_n$ and $v = v_n = \frac{1}{4}\Delta t u'(t_n)$ in the local error decomposition (5.39). The vector ξ will satisfy this relation if $(I + \frac{1}{4}I_2 Z)\xi = I_1 v$, that is

$$I_1\xi = I_1 v\,, \qquad I_2\big(I + \frac{1}{4}Z\big)\xi = 0\,.$$

It is seen that $\xi = [\xi_j] \in \mathbb{R}^m$ is given by

$$\xi_j = v_j \quad (\text{for } j < i)\,, \qquad \xi_{i+k} = \Big(\frac{\nu_j}{\nu_j - 4}\Big)^{k+1} v_{i-1} \quad (\text{for } k \ge 0)\,,$$

where $\nu_j = \Delta t/\Delta x_j$. Therefore $\|\xi\|_\infty \le \|v\|_\infty$ if $\nu_j \le 2$ on $\mathcal{I}_2$.

It follows that for this linear advection case, the local error decomposition (5.39) will be valid under (5.31), with $\|\xi_n\|_\infty = \mathcal{O}(\Delta t)$, $\|\xi_{n+1} - \xi_n\|_\infty = \mathcal{O}(\Delta t^2)$, and with $\|\eta_n\|_\infty = \mathcal{O}(\Delta t^2)$ containing the higher-order terms in the local error, leading to convergence with order one.

5.4 Second-order schemes

In the literature, several second-order multirate schemes for conservation laws have been derived that are based on the standard two-stage Runge-Kutta method

$$u^*_{n+1} = u_n + \Delta t F(u_n)\,, \qquad u_{n+1} = u_n + \tfrac{1}{2}\Delta t\big(F(u_n) + F(u^*_{n+1})\big)\,.$$

The second stage can also be written as

$$u_{n+1} = \frac{1}{2}u_n + \frac{1}{2}\big(u^*_{n+1} + \Delta t F(u^*_{n+1})\big)\,.$$

Monotonicity properties are more clear with this form. The method is known as the explicit trapezoidal rule or the modified Euler method. In this section we consider some multirate schemes, based on this method, with one level of temporal refinement. Results on internal consistency and mass conservation are mentioned here, but a detailed discussion will only be given in Section 5.5.

The second-order scheme of Tang & Warnecke [54] reads

$$\begin{cases} u^*_{n+\frac{1}{2}} = u_n + \frac{1}{2}\Delta t F(u_n)\,, \\ u_{n+\frac{1}{2}} = \frac{1}{2}\big(u_n + u^*_{n+\frac{1}{2}} + \frac{1}{2}\Delta t F(u^*_{n+\frac{1}{2}})\big)\,, \\ u^*_{n+1} = I_1\big(u_n + \Delta t F(u_n)\big) + I_2\big(u_{n+\frac{1}{2}} + \frac{1}{2}\Delta t F(u_{n+\frac{1}{2}})\big)\,, \\ u_{n+1} = \frac{1}{2}I_1\big(u_n + u^*_{n+1} + \Delta t F(u^*_{n+1})\big) + \frac{1}{2}I_2\big(u_{n+\frac{1}{2}} + u^*_{n+1} + \frac{1}{2}\Delta t F(u^*_{n+1})\big)\,. \end{cases} \tag{5.42}$$

We will refer to this scheme as TW2. It will be shown below that this scheme is internally consistent but not mass-conserving.

Constantinescu & Sandu [8] introduced the following scheme, which will be referred to as CS2,

$$\begin{cases} u^*_{n+\frac{1}{2}} = u_n + \Delta t I_1 F(u_n) + \frac{1}{2}\Delta t I_2 F(u_n)\,, \\ u_{n+\frac{1}{2}} = u_n + \frac{1}{4}\Delta t I_2\big(F(u_n) + F(u^*_{n+\frac{1}{2}})\big)\,, \\ u^*_{n+1} = I_1\big(u_n + \Delta t I_1 F(u_{n+\frac{1}{2}})\big) + I_2\big(u_{n+\frac{1}{2}} + \frac{1}{2}F(u_{n+\frac{1}{2}})\big)\,, \\ u_{n+1} = u_n + \frac{1}{4}\Delta t\big(F(u_n) + F(u^*_{n+\frac{1}{2}}) + F(u_{n+\frac{1}{2}}) + F(u^*_{n+1})\big)\,. \end{cases} \tag{5.43}$$

This scheme is mass-conserving but not internally consistent. Nevertheless, we will see that it is still convergent (with order one) in the maximum-norm due to damping and cancellation effects. Note that for non-stiff ODE systems the scheme will be consistent and convergent with order two.

The related method of Dawson and Kirby [9] is also mass-conserving but not internally consistent. However in that scheme a limiter is applied which is adapted to the outcome of previous stages, so it does not fit in the framework of this chapter where the semi-discrete system is supposed to be given a priori.

In Savcenco [45] several other multirate schemes of order two can be found for stiff (parabolic) problems. These are Rosenbrock-type schemes that contain a parameter γ, and setting $\gamma = 0$ yields an explicit scheme. We consider here the scheme that was introduced in [47]; it will be referred to as SHV2. In this scheme, first a prediction $\bar{u}_{n+1}$ is computed, followed by refinement steps on $\mathcal{I}_2$

using interpolated values $\bar{u}_{n+1/2}$ on $\mathcal{I}_1$. The scheme reads

$$\begin{cases} \bar{u}^*_{n+1} = u_n + \Delta t F(u_n)\,, \\ \bar{u}_{n+1} = \frac{1}{2}u_n + \frac{1}{2}\bar{u}^*_{n+1} + \frac{1}{2}\Delta t F(\bar{u}^*_{n+1})\,, \\ \bar{u}_{n+\frac{1}{2}} = \frac{1}{2}u_n + \frac{1}{4}\bar{u}_{n+1} + \frac{1}{4}\bar{u}^*_{n+1}\,, \\ u^*_{n+\frac{1}{2}} = I_1\bar{u}_{n+\frac{1}{2}} + I_2\big(u_n + \frac{1}{2}\Delta t F(u_n)\big)\,, \\ u_{n+\frac{1}{2}} = I_1\bar{u}_{n+\frac{1}{2}} + I_2\big(\frac{1}{2}u_n + \frac{1}{2}u^*_{n+\frac{1}{2}} + \frac{1}{4}\Delta t F(u^*_{n+\frac{1}{2}})\big)\,, \\ u^*_{n+1} = I_1\bar{u}_{n+1} + I_2\big(u_{n+\frac{1}{2}} + \frac{1}{2}\Delta t F(u_{n+\frac{1}{2}})\big)\,, \\ u_{n+1} = I_1\bar{u}_{n+1} + I_2\big(\frac{1}{2}u_{n+\frac{1}{2}} + \frac{1}{2}u^*_{n+1} + \frac{1}{4}\Delta t F(u^*_{n+1})\big)\,. \end{cases} \tag{5.44}$$

This scheme will be seen to be internally consistent but not mass-conserving. We note that (5.44) could be written with fewer stages; there are no function evaluations of $\bar{u}_{n+1}$ and $\bar{u}_{n+\frac{1}{2}}$, so these vectors are just included for notational convenience. Further we note that this scheme was not intended originally as used here. Instead, the prediction values $\bar{u}^*_{n+1}$ and $\bar{u}_{n+1}$ were used in [47] to estimate local errors, and based on this estimate the partitioning $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$ was adjusted. For the schemes in the present chapter the partitioning is supposed to be given, based on local Courant numbers.

The interpolation step in (5.44) can be written as

$$\bar{u}_{n+\frac{1}{2}} = \frac{3}{4}u_n + \frac{1}{4}\bar{u}_{n+1} + \frac{1}{4}\Delta t F(u_n)\,, \tag{5.45}$$

which corresponds to quadratic Hermite interpolation. As an alternative we can also consider linear interpolation

$$\bar{u}_{n+\frac{1}{2}} = \frac{1}{2}u_n + \frac{1}{2}\bar{u}_{n+1}\,, \tag{5.46}$$

but in the numerical tests (5.45) gave somewhat better results (errors approximately 5% smaller) in general.

In practical applications, for systems of conservation laws, evaluation of the function components $F_j(v)$ will be the main computational work. Note that if $I_k F(v)$ is needed then v should be known on $\mathcal{I}_k$ and on a few additional points near the interface (how many points depends on the stencil of the spatial discretization). If we ignore these interface points, and assume that $\mathcal{I}_k$ contains m_k points, $m_1 + m_2 = m$, then we can easily estimate the amount of work per step with the schemes. For the schemes TW2 and SHV2 this is $2(m + m_2)\mu_{\mathrm{W}}$, and for the CS2 scheme it is $4m\mu_{\mathrm{W}}$, where μ_{W} is the measure of work for a single component $F_j(v)$. Therefore, if $m_2 \ll m_1$, that is, temporal refinement is only needed at few points, then the CS2 scheme will be approximately twice as expensive as the other two schemes.

TABLE 5.1: Results for the smooth advection problem with the CS2, TW2 and SHV2 schemes. Maximum errors and L_1-errors at final time $t_N = T$ for various m with fixed Courant number $\nu = 0.4$.

m	100	200	400	800
CS2, $\|e_N\|_\infty$	$1.97 \cdot 10^{-3}$	$5.64 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$9.96 \cdot 10^{-5}$
CS2, $\|e_N\|_1$	$7.11 \cdot 10^{-4}$	$1.84 \cdot 10^{-4}$	$4.85 \cdot 10^{-5}$	$1.28 \cdot 10^{-5}$
TW2, $\|e_N\|_\infty$	$6.08 \cdot 10^{-4}$	$1.57 \cdot 10^{-4}$	$3.98 \cdot 10^{-5}$	$9.99 \cdot 10^{-6}$
TW2, $\|e_N\|_1$	$2.85 \cdot 10^{-4}$	$7.35 \cdot 10^{-5}$	$1.86 \cdot 10^{-5}$	$4.66 \cdot 10^{-6}$
SHV2, $\|e_N\|_\infty$	$6.10 \cdot 10^{-4}$	$1.57 \cdot 10^{-4}$	$3.95 \cdot 10^{-5}$	$9.90 \cdot 10^{-6}$
SHV2, $\|e_N\|_1$	$2.91 \cdot 10^{-4}$	$7.40 \cdot 10^{-5}$	$1.86 \cdot 10^{-5}$	$4.66 \cdot 10^{-6}$

5.4.1 Numerical tests

An analysis of the above second-order schemes will be given in the next section in the framework of partitioned Runge-Kutta methods. Here we already present some numerical results that will serve as benchmarks for the analysis.

Linear advection with smooth solution

As a first test on the accuracy of the schemes we consider the linear advection equation (5.1) on the spatial interval $0 < x < 1$ with periodic boundary conditions, and time interval $0 < t \leq T = 1$. For test purposes a uniform spatial grid is taken, so that interface effects are certainly not due to the spatial discretization, for which the WENO5 scheme is chosen; the formulas for this discretization can be found for example in [48]. Temporal refinement is used at the union of spatial intervals $\mathcal{D}_k = \{x : |x - k/10| \leq 1/40\}$, $k = 1, \ldots, 9$, and we consider a fixed Courant number $\nu = \Delta t/\Delta x = 0.4$.

For this accuracy test a smooth solution $u(x,t) = \sin^2(\pi(x-t))$ is considered. The errors in the maximum-norm and discrete L_1-norm ($\|v\|_1 = \sum_j \Delta x_j |v_j|$) are presented in Table 5.1. It is seen that with the CS2 scheme we have only first-order convergence in the maximum-norm, due to the interface points; the L_1-errors are still second-order. For the schemes TW2 and SHV2 we see an order two convergence also in the maximum-norm. The entries in Table 5.1 are the total (absolute) errors with respect to the PDE solution, but it was verified that the spatial errors are much smaller here than the temporal errors.

To see that the large errors for scheme CS2 in the maximum-norm are indeed caused by the interface points, the errors as function of x at the final time with $m = 800$ are displayed in Figure 5.1. The (relatively) large errors for CS2 at the interface points are clearly visible. For scheme TW2 there are no visible interface effects. The errors for SHV2 are almost the same as for TW2.

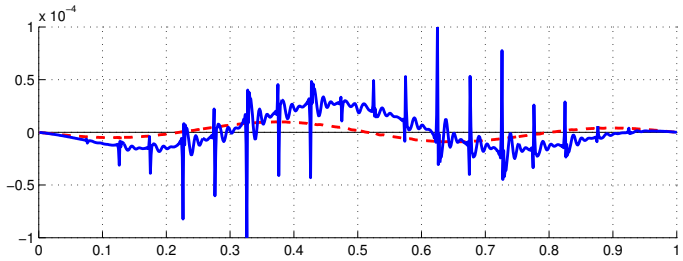


FIGURE 5.1: Errors versus $x_j \in (0,1)$ at final time $t_N = T$ for the schemes CS2 (thick solid line) and TW2 (thick dashed line), $m = 800$.

The CS2 scheme is not internally consistent at the interfaces, but we see in this test that it is still convergent. This is similar as with the OS1 scheme.

The linear advection test was repeated with an initial block-function with the aim of seeing the effect of the lack of mass-conservation for the TW2 and SHV2 schemes. In general, mass conservation is needed to guarantee a correct shock speed and shock location. However, this test with a block function showed very little difference between the schemes.

Burgers' equation with stationary shock

In the above numerical test the lack of mass conservation for scheme TW2 only gave a very small effect. To make this effect more pronounced we consider the Burgers equation with a stationary shock at a grid interface. The equation is given by

$$u_t + \frac{1}{2}\big(u^2\big)_x = 0 \tag{5.47}$$

for $0 < t < T = 0.3$ and $-1 < x < 1$, with initial profile

$$u(x,0) = \begin{cases} 1 & \text{if } |x| < 0.3\,, \\ -1 & \text{otherwise}\,, \end{cases}$$

and boundary conditions $u(-1,t) = u(1,t) = -1$. This will lead to a rarefaction wave around $x = -0.3$ and a stationary shock at $x = 0.3$. In this experiment refinement is used at $\mathcal{D} = \cup_{k=1}^{10}[y_k, y_k + 0.1]$, $y_k = 0.2\,k - 1.1$. So the stationary shock is located at a grid interface.

The spatial discretization is given by the limited TVD scheme of Appendix using a cell-centered non-uniform grid with mesh widths $\Delta x_j = \frac{1}{2}\Delta x$ if $x_j \in \mathcal{D}$, and $\Delta x_j = \Delta x$ otherwise. Also $\mathcal{I}_2 = \{j : x_j \in \mathcal{D}\}$ and $\mathcal{I}_1 = \mathcal{I} \setminus \mathcal{I}_2$, so that spatial and temporal refinements are taken at the same points.

Numerical solutions at the output time $t = T$ are shown in Figure 5.2 for $\Delta x = \frac{1}{80}$ and $\nu = \Delta t/\Delta x = 0.8$. The left picture shows the solution with $-1 < x < 1$ for the CS2 scheme. Differences between the schemes are not well visible on this scale. Therefore the right picture shows a zoom around

$x = 0.3$ for the schemes TW2, CS2 and SHV2. One sees that with CS2 the shock location is correct; there is some smearing due to numerical diffusion in the spatial discretization, but it is more or less symmetric around $x = 0.3$. The solution of TW2 is leaning too much to the left, and for SHV2 too much to the right. This due to the lack of (local) conservation.

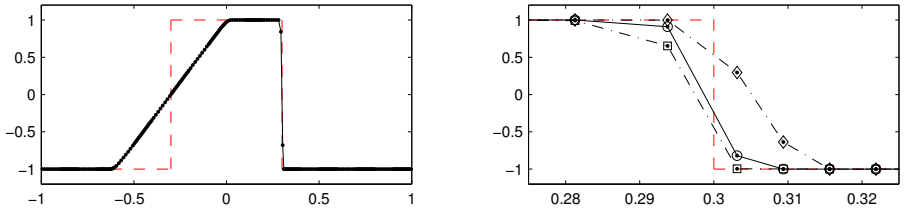


FIGURE 5.2: Numerical solutions at time $T = 0.3$ for $\Delta x = \frac{1}{80}$, $\nu = 0.8$. Left picture: initial profile (dashed), and semi-discrete solution for $-1 < x < 1$. Right picture: solutions around the stationary shock with the schemes TW2 (□ marks), CS2 (○ marks) and SHV2 (◇ marks), and with exact PDE solution (dashed line).

Let $M(v) = \sum_j \Delta x_j v_j$. (If the v_j were densities, this would be total mass; for Burgers' equation it is more natural to think of momenta.) Then $M(u(t_n)) - M(u_n)$ is a conservation defect. Figure 5.3 shows this defect at the final time $t_N = T$ for the three schemes on a fixed spatial mesh, $\Delta x = 1/160$, and with $\nu = \Delta t/\Delta x$ varying between 0 and 1.2. (We have taken $\nu = k/40$, $k = 1, 2, \ldots, 48$, with markers placed when ν is a multiple of 0.1.) In the same figure, middle plot, the increase of the total variation $\|u_N\|_{\mathrm{TV}}$ is displayed. The total variation should be 4, as for the PDE solution, and this is the numerical value for the semi-discrete system (within machine precision). In this example it is conserved with larger Courant numbers for the scheme CS2 than for TW2 and SHV2. The right plot in the figure shows the increase of the maximum norm $\|u_N\|_\infty - 1$.

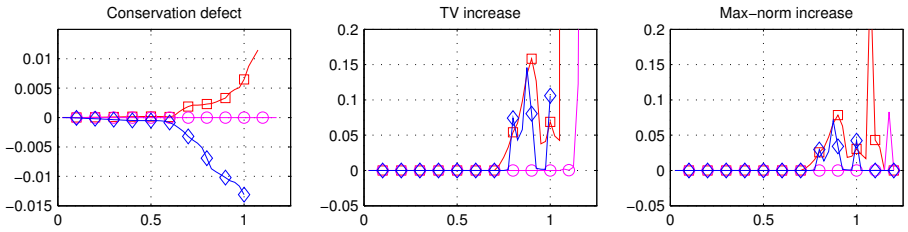


FIGURE 5.3: Conservation defects and increase of total variation and max-norm for $0 < \nu \leq 1.2$ with $\Delta x = \frac{1}{160}$, for the schemes TW2 (□ marks), CS2 (○ marks) and SHV2 (◇ marks).

In these figures overflow values are not plotted. The schemes CS2 remained stable in this test up to $\nu = 1.2$, which is slightly larger than with the other two schemes. The instabilities did emerge at the stationary shock. Adding some initial perturbations results in instability for $\nu > 1$ with all three schemes.

Finally, in Figure 5.4 the logarithm (base 10) of the L_1-errors of the three schemes are given, again for $\Delta x = 1/160$ with varying ν. Both the errors with respect to the semi-discrete solution and the errors with respect to the PDE solution are plotted. It is seen that the ODE errors for CS2 are smaller than for the other two schemes for large Courant numbers. That is due to the fact that CS2 has a smaller error near the stationary shock. However, this scheme is more inaccurate than TW2 and SHV2 in the rarefaction wave, similar as in the previous test, and that reveals itself in the larger error for small Courant numbers. In the PDE errors the spatial errors will become dominant for small time steps, so there the best results are found for CS2 overall. From the PDE point of view, temporal errors less then 10^{-3} are not relevant on this spatial grid where we have a spatial error of $3.4 \cdot 10^{-3}$ approximately (PDE error for $\nu \to 0$).

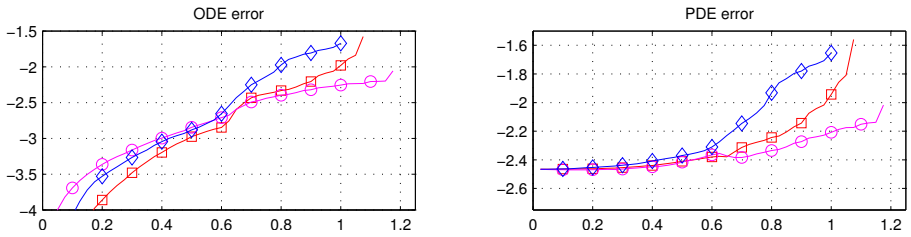


FIGURE 5.4: Logarithm ($\log_{10}$) of the L_1-errors, with respect to the exact semi-discrete solution (ODE error) and the exact PDE solution (PDE error), for $0 < \nu \leq 1.2$ with $\Delta x = \frac{1}{160}$. Results for the schemes TW2 ($\square$ marks), CS2 ($\circ$ marks) and SHV2 ($\diamond$ marks).

Burgers' equation with moving shock

The last test is again Burgers' equation (5.47), but now with a moving shock. We take $0 < t < T = 0.6$, $-1 < x < 1$ with initial profile

$$u(x,0) = \begin{cases} 1 & \text{if } -0.6 < x < 0\,, \\ 0 & \text{otherwise.} \end{cases}$$

and boundary conditions $u(-1,t) = u(1,t) = 0$. This will lead to a rarefaction wave between $x = -0.6 + t$ and $x = 0$, together with a moving shock at $x = \frac{1}{2}t$. Further, we use the same set-up as in the previous test.

The solutions at time $T = 0.6$ are shown in Figure 5.5. The enlargement around the shock at $x = 0.3$ now shows very little difference between the three

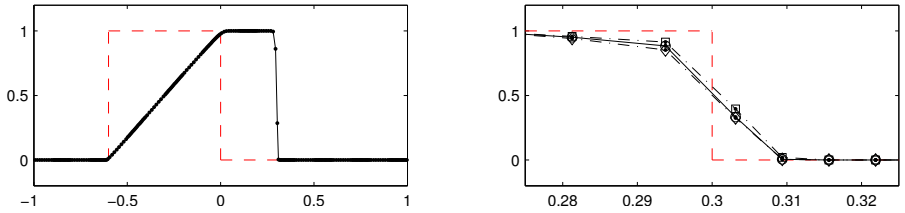


FIGURE 5.5: Numerical solutions at time $T = 0.6$ for $\Delta x = \frac{1}{80}$, $\nu = 0.8$. Left picture: initial profile (dashed), and semi-discrete solution for $-1 < x < 1$. Right picture: solutions around the moving shock with the schemes TW2 (□ marks), CS2 (○ marks) and SHV2 (◇ marks), and with exact PDE solution (dashed line).

schemes. So the lack of mass conservation for the TW2 and SHV2 schemes does not have much impact for this test. This is similar as in the tests of [54] for the TW2 scheme.

The conservation defects and the increase of total variation and maximum-norm, with fixed mesh width $\Delta x = \frac{1}{160}$ and variable ν, are displayed in Figure 5.6. Here we see that all three schemes start to loose the TVD property when Courant numbers become larger than 0.8, approximately. The plot on the right of the overshoot values $\|u_N\|_\infty - 1$ looks similar, except that now the increase starts at Courant number one. The loss of the TVD property for $\nu \in [0.8, 1]$ is cause by oscillations at the shock, not in the rarefaction wave.

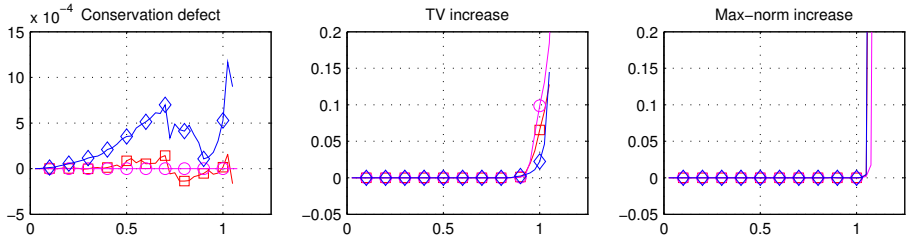


FIGURE 5.6: Conservation defects and increase of total variation and max-norm for $0 < \nu \leq 1.2$ with $\Delta x = \frac{1}{160}$, for the schemes TW2 (□ marks), CS2 (○ marks) and SHV2 (◇ marks).

We see that the conservation defect in this test is much smaller than in the previous test with a standing shock at a grid interface. Of course, both these tests are somewhat academic, but for practical situations the present test with a moving shock seems more relevant. Monotonicity for the TW2 and SHV2 schemes holds with larger Courant numbers than in the previous test. This is caused by the fact that in the previous test there were two incoming fluxes at the standing shock, whereas now we have one incoming and one outgoing

flux at each grid cell. In the standing shock test the conservation property of the CS2 scheme did suppress the tendency of increasing the total variation and maximum-norm.

In Figure 5.7 the temporal (ODE) errors and total (PDE) errors are plotted, again with fixed mesh width $\Delta x = \frac{1}{160}$ and variable ν. The ODE errors for the CS2 scheme are larger than for the other two schemes for small Courant numbers, but for the PDE errors this is not relevant here. In the plot of the PDE errors we see that here the SHV2 scheme gives somewhat larger errors than the TW2 and CS2 schemes. Detailed inspection of the solution plots revealed that this is due to a slight dissipation with SHV2 at the top and bottom of the rarefaction wave. We did notice, however, that these errors are quite sensitive to the precise set-up of the test. For example, with $T = 0.5$ and initial profile $u(0, x) = 1$ for $-T < x < 0$ and 0 otherwise, then the PDE errors of SHV2 were smaller than with the other two schemes for the larger Courant numbers.

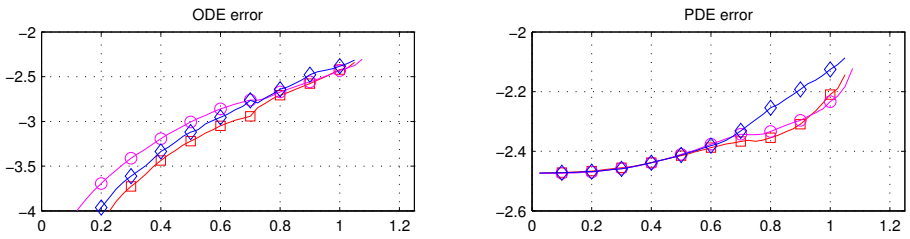


FIGURE 5.7: Logarithm ($\log_{10}$) of the L_1-errors, with respect to the exact semi-discrete solution (ODE error) and the exact PDE solution (PDE error), for $0 < \nu \leq 1.2$ with $\Delta x = \frac{1}{160}$. Results for the schemes TW2 (□ marks), CS2 (○ marks) and SHV2 (◇ marks).

For theoretical purposes it is interesting to note that with the Burgers flux function $f(u) = \frac{1}{2}u^2$ we have $f'(u) \in [0, 1]$ in this test. Furthermore, the mesh width in space is $\Delta x_j = \Delta x/k$ for $j \in \mathcal{I}_k$, $k = 1, 2$, and $\mu = 1$ for the used spatial discretization. Therefore, as discussed in Example 5.2.2, the monotonicity assumption (5.13) will be satisfied with

$$\tau_0 = \frac{1}{2}\Delta x$$

for both the maximum-norm and for the total variation semi-norm. Note that with the first-order upwind discretization this would be $\tau_0 = \Delta x$.

5.5 Partitioned Runge-Kutta methods

5.5.1 General properties

In the multirate examples considered thus far, only one level of refinement was used to keep the notation simple. Generalizations will be formulated in this

section in terms of partitioned Runge-Kutta methods; see also [8, 16]. This will enable us to present the schemes in a compact fashion. Since this chapter is concerned with schemes for conservation laws, we will restrict ourselves to explicit methods.

For the ODE system in $\mathbb{R}^m$, arising from semi-discretization of a PDE with given initial value,

$$u'(t) = F(u(t)), \quad u(0) = u_0, \tag{5.48}$$

let $\mathcal{I} = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_r$ be an index partitioning with corresponding diagonal matrices $I = I_1 + \cdots + I_r$, where the entries of the I_k are zero or one, and I is the identity matrix. For a time step from t_n to $t_{n+1} = t_n + \Delta t$, an explicit partitioned Runge-Kutta method reads

$$\begin{aligned} v_{n,i} &= u_n + \Delta t \sum_{k=1}^{r} \sum_{j=1}^{i-1} a_{ij}^{(k)} I_k F(v_{n,j}), \qquad i = 1, \ldots, s, \\ u_{n+1} &= u_n + \Delta t \sum_{k=1}^{r} \sum_{j=1}^{s} b_j^{(k)} I_k F(v_{n,j}). \end{aligned} \tag{5.49}$$

The internal stage vectors $v_{n,i}$, $i = 1, \ldots, s$, give approximations at intermediate time levels. The multirate schemes of the previous sections all fit in this form with $r = 2$. With $r > 2$ more levels of temporal refinement are allowed.

Internal consistency and conservation

Let $c_i^{(k)} = \sum_{j=1}^{i-1} a_{ij}^{(k)}$, $i = 1, \ldots, s$. If we have

$$c_i^{(k)} = c_i^{(l)} \quad \text{for all } 1 \le k, l \le r \text{ and } 1 \le i \le s, \tag{5.50}$$

then the internal vectors $v_{n,i}$ will be consistent approximations to $u(t_n + c_i \Delta t)$, and the method will be called *internally consistent*. As will be seen, this is an important property for the accuracy of the method when applied to semi-discrete systems.

Apart from consistency, we will also regard global *conservation*, for example mass conservation. Suppose that $h^T = [h_1, \ldots, h_m]$ is such that $h^T u(t) = \sum_j h_j u_j(t)$ is a conserved quantity for the ODE system (5.48). This will hold for arbitrary initial value u_0 provided that

$$h^T F(v) = 0 \qquad \text{for all } v \in \mathbb{R}^m. \tag{5.51}$$

For the partitioned Runge-Kutta scheme we have

$$\begin{aligned} h^T u_{n+1} &= h^T u_n + \Delta t \sum_{k=1}^{r} \sum_{j=1}^{s} b_j^{(k)} h^T I_k F(v_{n,j}) \\ &= h^T u_n + \Delta t \sum_{k \ne l} \sum_{j=1}^{s} \big(b_j^{(k)} - b_j^{(l)}\big) h^T I_k F(v_{n,j}), \end{aligned}$$

for any $1 \le l \le r$. Therefore, as noted in [8], the conservation property $h^T u_{n+1} = h^T u_n$ will be valid provided that

$$b_j^{(k)} = b_j^{(l)} \quad \text{for all } 1 \le k, l \le r \text{ and } 1 \le j \le s\,. \tag{5.52}$$

Order conditions for non-stiff problems

Below we shall use the order conditions for partitioned Runge-Kutta methods applied to non-stiff problems as found in [18, Thm. I.15.9] for $r = 2$. This classical order will be denoted by p. As we will see, it often does not correspond to the order of convergence for semi-discrete systems, and therefore p is often referred to as the *classical order*.

To write the order conditions in a compact way, let $A_k = [a_{ij}^{(k)}] \in \mathbb{R}^{s\times s}$ and $b_k = [b_i^{(k)}] \in \mathbb{R}^s$ contain the coefficients of the method, and set $e = [1, \dots, 1]^T \in \mathbb{R}^s$. The conditions for $p = 1$ are just

$$b_k^T e = 1 \qquad \text{for } \; k = 1, \dots, r\,, \tag{5.53}$$

that is $\sum_{j=1}^s b_j^{(k)} = 1$ for all k. To have $p = 2$ the coefficients should satisfy

$$b_k^T A_l\, e = \frac{1}{2} \qquad \text{for } \; k, l = 1, \dots, r\,. \tag{5.54}$$

The number of conditions quickly increase for higher orders; for $p = 3$ we get

$$b_k^T C_{l_1} A_{l_2} e = \frac{1}{3}\,, \quad b_k^T A_{l_1} A_{l_2} e = \frac{1}{6} \qquad \text{for } \; k, l_1, l_2 = 1, \dots, r\,, \tag{5.55}$$

where $C_l = \operatorname{diag}(A_l e)$.

Formulation for non-autonomous systems

For non-autonomous systems

$$u'(t) = F(t, u(t))\,, \qquad u(0) = u_0\,, \tag{5.56}$$

we will use the partitioned method (5.49) with the stage function values $F(v_{n,j})$ replaced by $F(t_n + c_j \Delta t, v_{n,j})$. If (5.50) is valid, the abscissa are naturally taken as $c_i = c_i^{(k)}$, which is independent of k.

If (5.50) does not hold, then a proper choice of the abscissa is less obvious. For the OS1 and CS2 multirate schemes with $r = 2$ it is natural to take $c_i = c_i^{(2)}$. As generalization we will therefore use

$$c_i = c_i^{(r)}\,, \qquad i = 1, \dots, s\,. \tag{5.57}$$

Note that if $h^T F(t, v) = 0$ for all $t \in \mathbb{R}$, $v \in \mathbb{R}^m$, then we still have the conservation property $h^T u_{n+1} = h^T u_n$ if the scheme satisfies (5.52).

The alternative of replacing $I_k F(v_{n,j})$ in (5.49) by $I_k F(t_n + c_j^{(k)} \Delta t, v_{n,j})$ will destroy this conservation property. If the non-autonomous form originates from a source term in the PDE, this loss of conservation may be of little concern, but for the advection equation $u_t + \big(a(x,t)u\big)_x = 0$ with time-dependent velocity it is still a very desirable property.

Example 5.5.1 The OS1 scheme (5.9) leads to the partitioned method (5.49) with $r = 2$ and coefficients given by

$$\begin{array}{c|c|c} a_{ij}^{(1)} & a_{ij}^{(2)} \\ \hline b_j^{(1)} & b_j^{(2)} \end{array} = \begin{array}{cc|cc} 0 & & 0 & \\ 0 & 0 & 1/2 & 0 \\ \hline 1/2 & 1/2 & 1/2 & 1/2 \end{array}$$

For non-autonomous systems $u'(t) = F(t, u(t))$ the scheme with (5.57) reads

$$\begin{cases} u_{n+\frac{1}{2}} = u_n + \frac{1}{2}\Delta t I_2 F(t_n, u_n)\,, \\ u_{n+1} = u_n + \frac{1}{2}\Delta t F(t_n, u_n) + \frac{1}{2}\Delta t F(t_{n+\frac{1}{2}}, u_{n+\frac{1}{2}})\,. \end{cases}$$

The use of $I_k F(t_n + c_j^{(k)}\Delta t, v_{n,j})$ instead of $I_k F(t_n + c_j\Delta t, v_{n,j})$, $c_j = c_j^{(2)}$, would lead to the same formula for $u_{n+1/2}$ in the first stage, but then

$$u_{n+1} = u_n + \frac{1}{2}\Delta t F(t_n, u_n) + \frac{1}{2}\Delta t I_1 F(t_n, u_{n+\frac{1}{2}}) + \frac{1}{2}\Delta t I_2 F(t_{n+\frac{1}{2}}, u_{n+\frac{1}{2}})\,,$$

which is no longer conservative. □

The above order conditions have been derived for autonomous systems, but with (5.57) they are also valid for non-autonomous systems. This follows from the fact that $u'(t) = F(t, u(t))$ can be written as an equivalent, augmented autonomous system $u'(t) = F(\vartheta(t), u(t))$, $\vartheta'(t) = 1$, with $\vartheta(0) = 0$, and application of the partitioned method to this augmented system gives the same result as to the original, non-autonomous system provided the additional equation $\vartheta'(t) = 1$ is included in the index set $\mathcal{I}_r$.

Conservation versus internal consistency

For the multirate schemes that have been considered in this chapter, the conditions for internal consistency (5.50) and conservation (5.52) did not match. This incompatibility is valid for all 'genuine' multirate schemes that are based on one single method $\mathcal{M}_{\mathrm{RK}}$, that is, for schemes (5.49) that reduce to m_k applications (with step size $\Delta t/m_k$) of this base method $\mathcal{M}_{\mathrm{RK}}$ to cover $[t_n, t_{n+1}]$ in case that $\mathcal{I}_k = \mathcal{I}$ and the other $\mathcal{I}_l$ are empty.

Consider, as simple example, a quadrature problem $u'(t) = g(t) \in \mathbb{R}^m$, which is just a special case of (5.56). (In a PDE context, this can be viewed as a degenerate case of advection with a source term where the advective velocity happens to be zero.) Suppose (5.52) is valid, and let $\mathcal{J} = \{i \in \mathcal{I} : b_i \neq 0\}$. Then for the quadrature problem we simply get

$$u_{n+1} = u_n + \Delta t \sum_{i \in \mathcal{J}} b_i\, g(t_n + c_i \Delta t)\,,$$

which is independent of the partitioning. However, if this is the result of a base method $\mathcal{M}_{\rm RK}$ with $m_1 = 1$, $\mathcal{I}_1 = \mathcal{I}$, then the result for $m_2 = 2$, $\mathcal{I}_2 = \mathcal{I}$ should be

$$u_{n+1} = u_n + \frac{1}{2}\Delta t \sum_{i\in\mathcal{J}} b_i \Big(g\big(t_n + \frac{1}{2}c_i\Delta t\big) + g\big(t_n + \frac{1}{2}(1+c_i)\Delta t\big)\Big)\,,$$

which is not the same for arbitrary source terms g.

Note that for general partitioned Runge-Kutta methods there is no conflict between (5.50) and (5.52). Given a scheme with the same $c_i^{(k)} = c_i^{(l)}$ (for all i,k,l), but different weights $b_i^{(k)} \neq b_i^{(l)}$ (for some i,k,l), we can add an extra stage with new weights b_i^* that are independent of k, to make it mass-conserving. Of course, this will increase the computational work per step, and for the TW1, TW2 and SHV2 schemes such a modification does not seem to lead to efficient schemes.

5.5.2 Monotonicity and convex Euler combinations

We are in particular interested in the case where the partitioned Runge-Kutta method (5.49) stands for a multirate scheme that takes m_k substeps of size $\Delta t/m_k$ on $\mathcal{I}_k$ to cover $[t_n, t_{n+1}]$, $k = 1,\dots,r$, with $m_1 = 1 < m_2 < \cdots < m_r$. The corresponding monotonicity assumption is

$$\Big\| v + \sum_{k=1}^{r} \frac{\tau_k}{m_k} I_k F(v) \Big\| \;\le\; \|v\| \qquad \text{for all } v\in\mathbb{R}^m \text{ and } \tau_k \le \tau_0,\ k = 1,\dots,r\,, \tag{5.58}$$

where $\|\cdot\|$ is a convex functional or (semi-)norm. For theoretical purposes we will also consider

$$\big\| v + \frac{\tau_0}{m_k} I_k F(v) \big\| \;\le\; \|v\| \qquad \text{for all } v\in\mathbb{R}^m \text{ and } k = 1,\dots,r\,. \tag{5.59}$$

Of course, (5.58) implies (5.59). On the other hand, if (5.59) is valid, then the inequality in (5.58) will hold under the step size restriction $\tau_1 + \cdots + \tau_m \le \tau_0$. If we are dealing with the maximum-norm, then (5.58) and (5.59) are equivalent.

In the following we denote for $l = 1,\dots,r$,

$$\begin{cases} \kappa_{ij}^{(l)} &= m_l a_{ij}^{(l)}\,, & 1 \le i,j \le s\,,\\ \kappa_{s+1,j}^{(l)} &= m_l b_j^{(l)}\,, & 1 \le j \le s\,,\\ \kappa_{i,s+1}^{(l)} &= 0\,, & 1 \le i \le s+1\,. \end{cases} \tag{5.60}$$

These coefficients will be grouped in the $(s+1)\times(s+1)$ matrix $\mathcal{K}_l = [\kappa_{ij}^{(l)}]$. It is convenient to add $v_{n,s+1} = u_{n+1}$ to the internal vectors. Then (5.49) can be written as

$$v_{n,i} \;=\; u_n + \sum_{l=1}^{r}\sum_{j=1}^{i-1} \kappa_{ij}^{(l)} \frac{\Delta t}{m_l} I_l F(v_{n,j})\,, \qquad i = 1,\dots,s+1\,. \tag{5.61}$$

Depending on the monotonicity assumption, we can consider various ways to represent this partitioned scheme in terms of convex Euler combinations. For this we will introduce new method coefficients $\alpha_{ij}^{(k)}$, $\beta_{ij}^{(k)}$ with corresponding lower triangular matrices $\mathcal{A}_k = [\alpha_{ij}^{(k)}]$ and $\mathcal{B}_k = [\beta_{ij}^{(k)}]$. Such convex Euler forms are also called Shu-Osher forms, after [49] where such representations were used originally to demonstrate the TVD property of certain Runge-Kutta methods.

Inequalities for matrices or vectors in this section are to be understood component-wise, that is, $P = [p_{ij}] \geq 0$ means that all p_{ij} are non-negative. Furthermore, if $P \in \mathbb{R}^{(s+1)\times q_1}$ and $Q \in \mathbb{R}^{(s+1)\times q_2}$, then $[P\ Q]$ stands for the matrix whose first q_1 columns equal those of P and the other columns equal those of Q. In this section we let $e = [1, 1, \ldots, 1]^T \in \mathbb{R}^{s+1}$, and we use the convention $\alpha/\beta = +\infty$ if $\alpha \geq 0$, $\beta = 0$.

Convex Euler form I: maximum-norm monotonicity.

A suitable form of (5.61) to obtain results on monotonicity in the maximum-norm is

$$v_{n,i} \;=\; \sum_{k=1}^{r} I_k\Big(\big(1-\alpha_i^{(k)}\big)u_n + \sum_{j=1}^{i-1}\big(\alpha_{ij}^{(k)} v_{n,j} + \beta_{ij}^{(k)}\frac{\Delta t}{m_k}F(v_{n,j})\big)\Big)\,, \qquad (5.62)$$

where $\alpha_i^{(k)} = \sum_{j=1}^{i-1}\alpha_{ij}^{(k)}$ and $i = 1, \ldots, s+1$. To have correspondence between (5.61) and (5.62) the coefficients should satisfy

$$\mathcal{K}_k = \big(I - \mathcal{A}_k\big)^{-1}\mathcal{B}_k\,, \qquad k = 1, \ldots, r\,. \qquad (5.63)$$

Further we want the coefficients to be such that

$$\alpha_i^{(k)} \leq 1\,, \quad \alpha_{ij}^{(k)}, \beta_{ij}^{(k)} \geq 0 \qquad \text{for } 1 \leq j < i \leq s+1\,,\ 1 \leq k \leq r\,. \qquad (5.64)$$

For such coefficients, let

$$C \;=\; \min_{i,j,k}\, \alpha_{ij}^{(k)}/\beta_{ij}^{(k)}\,. \qquad (5.65)$$

If there are no coefficients such that (5.63) and (5.64) are satisfied, we set $C = 0$.

Theorem 5.5.1 *Consider (5.62) with (5.64) and let C be given by (5.65). Assume (5.58) is valid in the maximum-norm. Then $\|u_{n+1}\|_\infty \leq \|u_n\|_\infty$ whenever $\Delta t \leq C\tau_0$.*

Proof. The form (5.62) is equivalent to

$$I_k v_{n,i} \;=\; I_k\Big(\big(1-\alpha_i^{(k)}\big)u_n + \sum_{j=1}^{i-1}\big(\alpha_{ij}^{(k)} v_{n,j} + \beta_{ij}^{(k)}\frac{\Delta t}{m_k}I_k F(v_{n,j})\big)\Big)\,, \qquad k = 1, \ldots, r\,.$$

We have $v_{n,1} = u_n$. Suppose (induction assumption) that $\|v_{n,j}\|_\infty \leq \|u_n\|_\infty$ for $j = 1, \ldots, i-1$. Since

$$\alpha_{ij}^{(k)} v_{n,j} + \beta_{ij}^{(k)}\frac{\Delta t}{m_k}I_k F(v_{n,j}) = \big(\alpha_{ij}^{(k)} - C\beta_{ij}^{(k)}\big)v_{n,j} + C\beta_{ij}^{(k)}\big(v_{n,j} + \frac{\Delta t}{C m_k}I_k F(v_{n,j})\big)\,,$$

we then have

$$\|\alpha_{ij}^{(k)} v_{n,j} + \beta_{ij}^{(k)} \frac{\Delta t}{m_k} I_k F(v_{n,j})\|_\infty \le \alpha_{ij}^{(k)} \|v_{n,j}\|_\infty \le \alpha_{ij}^{(k)} \|u_n\|_\infty \,.$$

It follows that $\|I_k v_{n,i}\|_\infty \le \|u_n\|_\infty$ for $k = 1, \ldots, r$, and hence $\|v_{n,i}\|_\infty \le \|u_n\|_\infty$. Using induction with respect to $i = 1, \ldots, s+1$ the proof thus follows. □

It is obvious that we are in particular interested in the optimal value of C in (5.65) for a given method (5.61). To obtain a suitable expression for this optimal value, we can follow the construction of Ferracina & Spijker [17] and Higueras [21] for the individual Runge-Kutta methods given by the coefficients $\mathcal{K}_k$.

Theorem 5.5.2 *The optimal value for $C \ge 0$ in (5.65), under the constraints (5.63) and (5.64), equals the largest $\gamma \ge 0$ such that*

$$(I + \gamma \mathcal{K}_k)^{-1} [\![e \; \gamma \mathcal{K}_k]\!] \ge 0 \,, \qquad k = 1, \ldots, r \,. \tag{5.66}$$

Proof. Suppose $\gamma \ge 0$ is such that (5.66) holds. We take $\mathcal{B}_k = (I + \gamma \mathcal{K}_k)^{-1} \mathcal{K}_k$ and $\mathcal{A}_k = \gamma \mathcal{B}_k$. With this choice it is easily seen that (5.63) and (5.64) are valid and that (5.65) holds with $C = \gamma$.

On the other hand, suppose that we have (5.63), (5.64) and (5.65) with $C \ge 0$, and set $\gamma = C$. Then

$$\big(I + \gamma \mathcal{K}_k\big)^{-1} [\![e \; \gamma \mathcal{K}_k]\!] = \big(I - \mathcal{M}_k\big)^{-1} [\![(I - \mathcal{A}_k) e \;\; \gamma \mathcal{B}_k]\!] \,,$$

where $\mathcal{M}_k = \mathcal{A}_k - \gamma \mathcal{B}_k$. From (5.65) we know that $\mathcal{M}_k \ge 0$, and since it is a strictly lower triangular matrix we also have

$$(I - \mathcal{M}_k)^{-1} = I + \mathcal{M}_k + \mathcal{M}_k^2 + \ldots + \mathcal{M}_k^s \;\ge\; 0 \,.$$

It follows that (5.66) is valid. □

Convex Euler form II: monotonicity under (5.59)

If we assume (5.59) for a general (semi-)norm or convex functional, then a suitable form for (5.61) is

$$v_{n,i} \;=\; \big(1 - \underline{\alpha}_i^{(0)}\big) u_n + \sum_{k=1}^{r} \sum_{j=1}^{i-1} \big(\underline{\alpha}_{ij}^{(k)} v_{n,j} + \underline{\beta}_{ij}^{(k)} \frac{\Delta t}{m_k} I_k F(v_{n,j}) \big) \,, \tag{5.67}$$

where $\underline{\alpha}_i^{(0)} = \sum_{j=1}^{i-1} \big(\underline{\alpha}_{ij}^{(1)} + \cdots + \underline{\alpha}_{ij}^{(r)} \big)$, $i = 1, \ldots, s+1$, and

$$\mathcal{K}_k = \Big(I - \sum_{l=1}^{r} \underline{\mathcal{A}}_l \Big)^{-1} \underline{\mathcal{B}}_k \,, \qquad k = 1, \ldots, r \,. \tag{5.68}$$

We want

$$\underline{\alpha}_i^{(0)} \le 1\,, \quad \underline{\alpha}_{ij}^{(k)}, \underline{\beta}_{ij}^{(k)} \ge 0 \qquad \text{for } 1 \le j < i \le s+1\,,\ 1 \le k \le r\,, \tag{5.69}$$

with an optimal

$$\underline{C} = \min_{i,j,k} \underline{\alpha}_{ij}^{(k)} / \underline{\beta}_{ij}^{(k)}\,. \tag{5.70}$$

Theorem 5.5.3 *Assume (5.59) is valid.*
(i) Consider (5.67) with (5.69) and let $\underline{C}$ be given by (5.70). Then $\|u_{n+1}\| \le \|u_n\|$ whenever $\Delta t \le \underline{C}\tau_0$.
(ii) The optimal $\underline{C} \ge 0$ in (5.70), under the constraints (5.68) and (5.69), equals the largest $\gamma \ge 0$ such that

$$\Big(I + \sum_{l=1}^{r} \gamma \mathcal{K}_l\Big)^{-1} [e\ \gamma \mathcal{K}_k] \ge 0\,, \qquad k = 1, \ldots, r\,. \tag{5.71}$$

The proof of this result is similar to that of the Theorems 5.5.1 and 5.5.2. In fact, the result for $r = 2$ can be obtained directly from Higueras [22] and Spijker [52]. Further we note that the coefficient matrices $\underline{\mathcal{A}}_k$ and $\underline{\mathcal{B}}_k$ which lead to an optimal value $\underline{C}$ are in this case given by $\underline{\mathcal{B}}_k = (I + \sum_l \gamma \mathcal{K}_l)^{-1} \mathcal{K}_k$ and $\underline{\mathcal{A}}_k = \gamma \underline{\mathcal{B}}_k$.

Convex Euler form III: TVD property and monotonicity under (5.58)

Finally, if (5.58) is assumed for a general (semi-)norm or convex functional, then we consider

$$v_{n,i} \;=\; \big(1 - \overline{\alpha}_i^{(0)}\big) u_n + \sum_{j=1}^{i-1} \Big(\overline{\alpha}_{ij}^{(0)} v_{n,j} + \sum_{k=1}^{r} \overline{\beta}_{ij}^{(k)} \frac{\Delta t}{m_k} I_k F(v_{n,j}) \Big)\,, \tag{5.72}$$

where $\overline{\alpha}_i^{(0)} = \sum_{j=1}^{i-1} \overline{\alpha}_{ij}^{(0)}$, $i = 1, \ldots, s+1$, and

$$\mathcal{K}_k = (I - A_0)^{-1} B_k\,, \qquad k = 1, \ldots, r\,. \tag{5.73}$$

Here we want

$$\overline{\alpha}_i^{(0)} \le 1\,, \quad \overline{\alpha}_{ij}^{(0)}, \overline{\beta}_{ij}^{(k)} \ge 0 \qquad \text{for } 1 \le j < i \le s+1\,,\ 1 \le k \le r\,. \tag{5.74}$$

such that

$$\overline{C} = \min_{i,j,k} \overline{\alpha}_{ij}^{(0)} / \overline{\beta}_{ij}^{(k)} \tag{5.75}$$

is optimal.

Theorem 5.5.4 *Consider (5.72) with (5.74) and let $\overline{C}$ be given by (5.75). Assume (5.58) is valid. Then $\|u_{n+1}\| \le \|u_n\|$ whenever $\Delta t \le \overline{C}\tau_0$.*

The proof is similar to that of Theorem 5.5.1. For this case there is no convenient representation of the optimal $\overline{C}$. An optimization code can be used to determine this optimal value. However, from the previous results we obtain useful upper and lower bounds for $\overline{C}$.

Theorem 5.5.5 *The optimal values C, $\underline{C}$, $\overline{C}$ in (5.65), (5.70) and (5.75) satisfy*

$$\frac{1}{r}\overline{C} \;\le\; \underline{C} \;\le\; \overline{C} \;\le\; C\,.$$

Consequently, if $\underline{C} = 0$ then $\overline{C} = 0$.

Proof. Given an optimal $\overline{C}$ with corresponding coefficient matrices A_0, B_k, we can take $\mathcal{A}_k = A_0$, $\mathcal{B}_k = B_k$. Then (5.63) and (5.64) hold and $\min_{i,j,k} \alpha_{ij}^{(k)}/\beta_{ij}^{(k)} \ge \overline{C}$. Consequently we have $C \ge \overline{C}$ for the optimal value C.

Likewise, for a given optimal $\underline{C}$ with corresponding $\underline{\mathcal{A}}_k$, $\underline{\mathcal{B}}_k$, we can choose $B_k = \underline{\mathcal{B}}_k$, $A_0 = \sum_{l=1}^{r} \underline{\mathcal{A}}_l$. Then (5.73) and (5.74) hold and we have

$$\min_{i,j,k} \overline{\alpha}_{ij}^{(0)}/\overline{\beta}_{ij}^{(k)} \ge \underline{C},$$

showing that $\overline{C} \ge \underline{C}$.

On the other hand, for given optimal $\overline{C}$ with corresponding A_0, B_k, we can take $\underline{\mathcal{B}}_k = B_k$, $\underline{\mathcal{A}}_k = \frac{1}{r}A_0$. It follows that $\underline{C} \ge \frac{1}{r}\overline{C}$. □

Results for the multirate schemes with one level of refinement

The monotonicity results for the multirate schemes of the previous sections are presented in Table 5.2. The table gives the threshold values C, $\overline{C}$ and $\underline{C}$ for the various schemes. The results for the first-order schemes OS1 and TW1 can be derived analytically as in Section 5.3.1; we get $C = 1$, $\overline{C} = 2/3$, $\underline{C} = 1 - 1/\sqrt{3}$ for OS1, and $C = 1$, $\overline{C} = 2 - \sqrt{2}$, $\underline{C} = 1 - 1/\sqrt{3}$ for TW1. The threshold values C, $\overline{C}$ for the second-order schemes have been found numerically, using (5.66) and (5.71). For the TW2 and CS2 schemes we have $\underline{C} = 0$ and therefore also $\overline{C} = 0$. (The fact that $\underline{C} = 0$ for these two schemes can also be shown analytically, similar to [22], by considering (5.71) for small $\gamma > 0$.) The value of $\overline{C}$ for SHV2 was obtained with the MATLAB optimization code FMINIMAX. This does not provide a guarantee that the solution is a global optimum, and therefore this $\overline{C}$ is to be considered as a lower bound. The fact that we merely have $C = 1/2$ for the SHV2 scheme is due to the first stage. Finally we note that for the variant of that scheme with linear interpolation (5.46), instead of (5.45), it was found that $C = 1/2$, $\underline{C} = 0.304$, and the optimization code produced the same value $\overline{C} = 0.304$ for this variant.

As noted before, the result $C = 1$ for the OS1 and TW1 scheme was already given in [30, 37, 54] in terms of maximum principles. For the CS2 scheme the same result has been proved in [8].

Recall that the threshold values C are such that we will have monotonicity in the maximum-norm, as well as maximum principles, provided that $\Delta t \le C\tau_0$.

TABLE 5.2: Threshold values for the multirate schemes with one level of refinement. The entry $\overline{C}$ for the scheme SHV2 is a lower bound.

	C	$\overline{C}$	$\underline{C}$
OS1	1	0.667	0.423
TW1	1	0.580	0.423
TW2	1	0	0
CS2	1	0	0
SHV2	0.5	0.284	0.284

Likewise, for spatial discretization with limiting the TVD property will hold if $\Delta t \leq \overline{C}\tau_0$. All this under corresponding assumptions (5.13) for the semi-discrete system.

Comparison of these theoretical values with the experiments of Section 5.4.1 for Burgers' equation with the TW2, CS2 and SHV2 schemes does not show a clear correspondence. As was noted, in those experiments we had $\tau_0 = \frac{1}{2}\Delta x$ for both the maximum-norm and the total variation semi-norm. Therefore, with $\nu = \Delta t/\Delta x$, the TVD property is guaranteed by the above results for $\nu \leq \frac{1}{2}\overline{C}$ and the maximum principle for $\nu \leq \frac{1}{2}C$. For the Burgers' experiment with a moving shock it was noticed that for the schemes TW2, CS2 and SHV2 we had no overshoots for $\nu \leq 1$, whereas the TVD property was valid for $\nu \leq 0.8$ approximately. Therefore, for that test, the theoretical threshold values $\overline{C} = 0$ for the TW2 and CS2 schemes in Table 5.2 are much too pessimistic. The same seems to hold for the small value $C = \frac{1}{2}$ of the SHV2 scheme compared to the value $C = 1$ for TW2 and CS2. This may be caused by the fact that spatial discretizations with flux-limiting (or of WENO type) do add some local diffusion near very steep gradients, which may counteract an overshoot or increase of total variation of the time stepping scheme. However, for the discrepancy in the TVD results it is more likely that a more refined theory is needed. As noted before, it was shown in [30] that the OS1 scheme is TVD for a class of limited discretizations under the same step size restriction as for the maximum principle, but that proof does not lend itself to generalization for the higher-order schemes.

Remark 5.5.1 Refined TVD results for the OS1 and TW1 scheme were also discussed in Section 5.3.1. It was shown that the TVD thresholds of both the OS1 and TW1 schemes become 1 for the system (5.22) arising from linear advection with first-order upwind discretization in space.

Experimentally, using various partitionings, including random partitionings, we observed that for this system the thresholds for monotonicity in the maximum-norm are 1 for the TW2 and CS2 schemes, and approximately 0.66 for the SHV2 scheme, whereas the thresholds for the TVD property are 0.5 for

the TW2 and CS2 schemes, and 0.86 for the SHV2 scheme.

Furthermore, it should be noticed that having a bound $\|S\|_\infty \le 1$ for the amplification matrix S guarantees stability in the maximum norm for this linear problem, but this is not a necessary condition. The spectral radius of S was found to be bounded by 1 for Courant numbers $\nu_j = \Delta t/\Delta x_j \le k$ for $j \in \mathcal{I}_k$, $k = 1, 2$, for these three schemes, that is, including the SHV2 scheme. Note that having spectral radius bounded by 1 is of course necessary for stability, but it is not sufficient, not even in the L_2 norm because the amplification matrices S are not normal. □

5.5.3 Convergence for smooth problems

In this section we derive bounds for the discretization errors that are valid for semi-discrete hyperbolic systems with smooth solutions. The classical, non-stiff order conditions are then no longer sufficient to obtain convergence of order p, due to the fact that F contains negative powers of the mesh widths Δx_j in space. We will accept a restriction on $\Delta t/\Delta x_j$ but the resulting error bounds should not contain negative powers of Δx_j.

It is useful here to take also non-autonomous equations (5.56) into consideration. Then linear constant coefficient problems $u'(t) = Au(t) + g(t)$ with time dependent source terms are included. Such $g(t)$ may originate from a genuine source term in the PDE or from an inhomogeneous boundary condition.

To ensure stability, it will be assumed that

$$\big\|\tilde v - v + \frac{\tau_0}{m_k} I_k\big(F(t,\tilde v) - F(t,v)\big)\big\|_\infty \;\le\; \|\tilde v - v\|_\infty\,, \qquad k = 1,\dots,r\,, \quad (5.76)$$

for any two vectors $\tilde v, v \in \mathbb{R}^m$ and $t \in \mathbb{R}$. In applications to semi-discrete systems obtained from conservation laws this τ_0 will be proportional to the mesh widths used in the spatial discretization, and hence an upper bound $\Delta t \le C\tau_0$ on the step size will be a CFL restriction.

Perturbed schemes

Consider, along with (5.49) in non-autonomous form, the perturbed scheme

$$\begin{aligned}
\tilde v_{n,i} &= \tilde u_n + \Delta t \sum_{k=1}^{r}\sum_{j=1}^{i-1} a_{ij}^{(k)} I_k F(t_{n,j}, \tilde v_{n,j}) + \rho_{n,i}\,, \qquad i = 1,\dots,s\,,\\
\tilde u_{n+1} &= \tilde u_n + \Delta t \sum_{k=1}^{r}\sum_{j=1}^{s} b_{j}^{(k)} I_k F(t_{n,j}, \tilde v_{n,j}) + \sigma_n\,,
\end{aligned} \quad (5.77)$$

where $t_{n,j} = t_n + c_j\Delta t$ and the $\rho_{n,i}$, σ_n are perturbations. These perturbations will be used later on to obtain expressions for the discretization errors. In order to distinguish the accuracy of the u_n from those of the internal stages we will mainly use the standard form (5.49) rather than (5.61).

As before, let the matrices $A_k = [a_{ij}^{(k)}] \in \mathbb{R}^{s\times s}$ and the vectors $b_k = [b_i^{(k)}] \in \mathbb{R}^s$ contain the coefficients of the scheme. Further, for the vector of abscissa

$c = [c_i] \in \mathbb{R}^s$ we denote $c^j = [c_i^j]$ for $j \geq 1$, with $c^0 = e = [1, \ldots, 1]^T \in \mathbb{R}^s$. To make the dimensions fitting we will use the Kronecker products $\boldsymbol{A}_k = A_k \otimes I$, $\boldsymbol{b}_k^T = b_k^T \otimes I$, $\boldsymbol{c^j} = c^j \otimes I$ and $\boldsymbol{e} = e \otimes I$ with $m \times m$ identity matrix $I = I_{m\times m}$. Likewise, $\boldsymbol{I}_k = I \otimes I_k$ with $s \times s$ identity matrix $I = I_{s\times s}$. To make the notation consistent, the $ms \times ms$ identity matrix is denoted by $\boldsymbol{I}$.

Let $\boldsymbol{Z}_n = \mathrm{diag}(Z_{n,i}) \in \mathbb{R}^{ms\times ms}$ with

$$Z_{n,i}(\tilde{v}_{n,i} - v_{n,i}) = \Delta t\big(F(t_{n,i}, \tilde{v}_{n,i}) - F(t_{n,i}, v_{n,i})\big)\,. \tag{5.78}$$

In view of (5.76) these $Z_{n,i} \in \mathbb{R}^{m\times m}$ can be taken such that[2]

$$\big\|I + \frac{1}{\gamma m_k} I_k Z_{n,i}\big\|_\infty \;\leq\; 1 \qquad \text{for } \Delta t \leq \gamma\tau_0\,,\ \gamma > 0\,,\ k = 1, \ldots, r\,. \tag{5.79}$$

To write the difference of (5.77) and (5.49) in a compact form, let also $\boldsymbol{\rho}_n = [\rho_{n,i}] \in \mathbb{R}^{sm}$ and $\boldsymbol{v}_n = [v_{n,i}]$, $\tilde{\boldsymbol{v}}_n = [\tilde{v}_{n,i}] \in \mathbb{R}^{sm}$. Then

$$\begin{aligned} \tilde{\boldsymbol{v}}_n - \boldsymbol{v}_n &= \boldsymbol{e}(\tilde{u}_n - u_n) + \sum_{k=1}^{r} \boldsymbol{A}_k \boldsymbol{I}_k \boldsymbol{Z}_n(\tilde{\boldsymbol{v}}_n - \boldsymbol{v}_n) + \boldsymbol{\rho}_n\,, \\ \tilde{u}_{n+1} - u_{n+1} &= \tilde{u}_n - u_n + \sum_{k=1}^{r} \boldsymbol{b}_k^T \boldsymbol{I}_k \boldsymbol{Z}_n(\tilde{\boldsymbol{v}}_n - \boldsymbol{v}_n) + \sigma_n\,. \end{aligned} \tag{5.80}$$

Elimination of $\tilde{\boldsymbol{v}}_n - \boldsymbol{v}_n$ thus leads to

$$\tilde{u}_{n+1} - u_{n+1} = S_n(\tilde{u}_n - u_n) + \boldsymbol{r}_n^T \boldsymbol{\rho}_n + \sigma_n\,, \tag{5.81}$$

where

$$S_n = I + \boldsymbol{r}_n^T \boldsymbol{e}\,, \qquad \boldsymbol{r}_n^T = \Big(\sum_{k=1}^{r} \boldsymbol{b}_k^T \boldsymbol{I}_k \boldsymbol{Z}_n\Big)\Big(\boldsymbol{I} - \sum_{k=1}^{r} \boldsymbol{A}_k \boldsymbol{I}_k \boldsymbol{Z}_n\Big)^{-1}. \tag{5.82}$$

The following result provides stability for this recursion with a step size restriction $\Delta t \leq C\tau_0$, where C is the threshold for monotonicity in the maximum-norm. We can consider arbitrary matrices $\boldsymbol{Z}_n$ with blocks satisfying (5.79), so that these matrices are independent from the perturbations $\boldsymbol{\rho}_n$ and σ_n.

Lemma 5.5.1 *Consider (5.80). Assume (5.79) and $\Delta t \leq C\tau_0$. Then*

$$\|S_n\|_\infty \leq 1\,, \quad \|\boldsymbol{r}_n^T\|_\infty \leq 2s\,. \tag{5.83}$$

Proof. Denote $w_{n,i} = \tilde{v}_{n,i} - v_{n,i}$ and also $w_{n,s+1} = \tilde{u}_{n+1} - u_{n+1}$, $\rho_{n,s+1} = \sigma_n$. Then

$$w_{n,i} = \tilde{u}_n - u_n + \sum_{k=1}^{r} \sum_{j=1}^{i-1} \frac{1}{m_k} \kappa_{ij}^{(k)} I_k Z_{n,j} w_{n,j} + \rho_{n,i}\,, \qquad i = 1, \ldots, s+1\,.$$

[2] As noted before, if F is differentiable we can take the $Z_{n,i}$ as integrated Jacobian matrices, but also for non-differentiable F we can choose them to satisfy (5.78). This is similar to the fact that if $x, y \in \mathbb{R}^m$ with $\|y\|_\infty \leq \|x\|_\infty$, then there is an $V \in \mathbb{R}^{m\times m}$ such that $Vx = y$ and $\|V\|_\infty \leq 1$; for example, if $|x_k| = \|x\|_\infty$, the matrix with kth column $\frac{1}{x_k} y$ and the other columns zero.

Following the construction used in Theorem 5.5.2 with optimal coefficients $\beta_{ij}^{(k)} = \alpha_{ij}^{(k)}/\gamma$, $\gamma = C$, we obtain

$$I_k(w_{n,i}-\rho_{n,i}) = \big(1-\alpha_i^{(k)}\big) I_k(\tilde{u}_n-u_n)+\sum_{j=1}^{i-1}\alpha_{ij}^{(k)} I_k\Big(w_{n,j}+\frac{1}{\gamma m_k}Z_{n,j}w_{n,j}-\rho_{n,j}\Big)\,.$$

This leads to

$$\|I_k w_{n,i}\|_\infty - \|\rho_{n,i}\|_\infty \;\le\; \big(1-\alpha_i^{(k)}\big)\|\tilde{u}_n-u_n\|_\infty + \sum_{j=1}^{i-1}\alpha_{ij}^{(k)}\big(\|w_{n,j}\|_\infty+\|\rho_{n,j}\|_\infty\big)\,.$$

If we make the induction assumption

$$\|w_{n,j}\|_\infty \le \|\tilde{u}_n-u_n\|_\infty + L_j \max_{\iota\le j}\|\rho_{n,\iota}\|_\infty\,, \tag{5.84}$$

for $j = 1,\dots,i-1$, with $L_j = 2j-1$, then

$$\begin{aligned}\|I_k w_{n,i}\|_\infty \;&\le\; \|\tilde{u}_n-u_n\|_\infty + \sum_{j=1}^{i-1}\alpha_{ij}^{(k)}\big(L_j\max_{\iota\le j}\|\rho_{n,\iota}\|_\infty+\|\rho_{n,j}\|_\infty\big)+\|\rho_{n,i}\|_\infty\\ &\le\; \|\tilde{u}_n-u_n\|_\infty + (L_{i-1}+1)\max_{j\le i-1}\|\rho_{n,j}\|_\infty+\|\rho_{n,i}\|_\infty\,.\end{aligned}$$

Hence (5.84) will also be satisfied for $j = i$, and the proof thus follows. □

Note that without the internal perturbations we obtain a result on contractivity in the maximum-norm:

$$\|\tilde{u}_{n+1}-u_{n+1}\|_\infty \;\le\; \|\tilde{u}_n-u_n\|_\infty \qquad \text{whenever } \Delta t \le C\tau_0\,, \tag{5.85}$$

for any two parallel steps of the scheme (5.49), starting with $\tilde{u}_n$ and u_n, respectively. In the above proof, the arguments leading to monotonicity have been copied. A more elegant and direct way to deduce contractivity from monotonicity is found in [52, p. 1236], following a construction of [6] for inner-product norms.

Local and global discretization errors

Throughout this section we will denote by $\mathcal{O}(\Delta t^q)$ a term or vector that can be bounded in norm by $K\Delta t^q$, for $\Delta t > 0$ small enough, with K not depending on the mesh widths Δx_j in the spatial discretization. The norm in this section is the maximum-norm. Moreover it will be tacitly assumed that the exact solution is smooth, so that derivatives of $u(t)$ are $\mathcal{O}(1)$.

Let $e_n = u(t_n)-u_n$ be the global discretization error at time level t_n, $n \ge 0$. To obtain a recursion for these global errors we can employ the above perturbed scheme with $\tilde{u}_n = u(t_n)$ and $\tilde{v}_{n,i} = u(t_{n,i})$, $t_{n,i} = t_n + c_i\Delta t$, $i = 1,\dots,s$. This

choice for the $\tilde{v}_{n,i}$ defines the perturbations $\rho_{n,i}$ and σ_n. Assuming the exact solution u to be $l+1$ times differentiable, Taylor expansion directly leads to

$$\begin{aligned} \boldsymbol{\rho}_n &= \sum_{k=1}^{r} \sum_{j=1}^{l} \frac{\Delta t^j}{j!} \big(\boldsymbol{c}^j - j \boldsymbol{A}_k \boldsymbol{c}^{j-1} \big) I_k u^{(j)}(t_n) + \mathcal{O}(\Delta t^{l+1}) \,, \\ \sigma_n &= \sum_{k=1}^{r} \sum_{j=1}^{l} \frac{\Delta t^j}{j!} \big(I - j \boldsymbol{b}_k^T \boldsymbol{c}^{j-1} \big) I_k u^{(j)}(t_n) + \mathcal{O}(\Delta t^{l+1}) \,. \end{aligned} \tag{5.86}$$

It follows that the global errors $e_n = u(t_n) - u_n$ satisfy the recursion

$$e_{n+1} = S_n e_n + d_n \,, \qquad n \geq 0 \,, \tag{5.87}$$

with local discretization errors d_n given by

$$d_n = \boldsymbol{r}_n^T \boldsymbol{\rho}_n + \sigma_n \,, \tag{5.88}$$

and with $S_n \in \mathbb{R}^{m \times m}$, $\boldsymbol{r}_n^T \in \mathbb{R}^{m \times ms}$ given by (5.82).

Note that from $\|S_n\|_\infty \leq 1$ it follows directly that consistency of order q (i.e., $\|d_n\|_\infty = \mathcal{O}(\Delta t^{q+1})$) implies convergence of order q (i.e., $\|e_n\|_\infty = \mathcal{O}(\Delta t^q)$), but we will see that the order of convergence can also be one larger than the order of consistency.

Let us first consider methods with classical order $p \geq 1$ that are not internally consistent, that is, $A_k e \neq A_l e$ for some k, l. Then the leading term in the local error is

$$d_n = \Delta t \, \boldsymbol{r}_n^T \sum_{k=1}^{r} (\boldsymbol{c} - \boldsymbol{A}_k \boldsymbol{e}) I_k u'(t_n) + \mathcal{O}(\Delta t^2) \,. \tag{5.89}$$

This gives an $\mathcal{O}(\Delta t)$ local error bound, which is of course quite poor. After all, d_n is the error that results after one step if $e_n = 0$. However, as we will see below, it can lead to convergence of order one.

Next assume the internal consistency condition (5.50) is satisfied, that is $A_k e = A_l e$ for $1 \leq k, l \leq r$. If $p = 1$ it follows directly that $\|d_n\|_\infty = \mathcal{O}(\Delta t^2)$. If $p \geq 2$ the leading term in the local discretization errors is given by

$$d_n = \Delta t^2 \boldsymbol{r}_n^T \sum_{k=1}^{r} \big(\frac{1}{2} \boldsymbol{c}^2 - \boldsymbol{A}_k \boldsymbol{c} \big) I_k u''(t_n) + \mathcal{O}(\Delta t^3) \,. \tag{5.90}$$

This still gives only consistency of order one, that is, an error $\mathcal{O}(\Delta t^2)$ after one step, but we will discuss below damping and cancellation effects that can lead to convergence with order two in this case.

For problems that are (mildly) stiff, such as semi-discrete systems from hyperbolic equations, the above derivation shows that *order reduction* is to be expected. This order reduction will appear primarily at interface points on the spatial grid, where the grid-functions $I_k u^{(j)}(t)$ have jumps. This is similar to the situation for standard Runge-Kutta methods, where order reduction appears at boundaries if the boundary values are time-dependent; see for instance

the review with references in [27, Sect. II.2]. With the partitioned and multirate schemes, we are creating interfaces that act like (internal) boundaries with time-dependent boundary conditions.

Based on the local error behavior, one would expect convergence with order one for the TW2 and SHV2 schemes, and lack of convergence for the scheme CS2. This is not what was seen in the numerical test in Section 5.4.1 for advection with a smooth solution. To obtain the correct (observed) order of convergence $q = 1, 2$, we need to study the propagation of the leading term in the local error. We already saw that the global error can be of the same order Δt^q as the local error if we have a suitable decomposition $d_n = (S_n - I)\xi_n + \eta_n$. In fact, we only need to study the principle term of the local error. It will be assumed that there exist vectors $\xi_n \in \mathbb{R}^m$, $n \geq 0$, such that

$$\left.\begin{aligned} &\Big\| \big(\boldsymbol{r}_n^T \boldsymbol{e}\big)\xi_n \;-\; \Delta t^q \boldsymbol{r}_n^T \sum_{k=1}^{r} \frac{1}{q!} \big(\boldsymbol{c}^q - q\boldsymbol{A}_k \boldsymbol{c}^{q-1}\big) I_k u^{(q)}(t_n) \Big\|_\infty = \mathcal{O}(\Delta t^{q+1})\,, \\ &\|\xi_n\|_\infty = \mathcal{O}(\Delta t^q)\,, \qquad \|\xi_{n+1} - \xi_n\|_\infty = \mathcal{O}(\Delta t^{q+1})\,. \end{aligned}\right\} \tag{5.91}$$

Then, following the proof of Theorem 5.3.2, we directly arrive at the following result.

Proposition 5.5.1 *Assume that (5.76) is valid, and let p be the (classical) order of the partitioned Runge-Kutta method.*
(i) If $p = 1$ and (5.91) holds with $q = 1$, then the method is convergent with order one in the maximum-norm.
(ii) Suppose that $p \geq 2$ and the method is internally consistent. Then, if (5.91) holds with $q = 2$, the method is convergent with order two in the maximum-norm.

The above result has been called a proposition, rather than a theorem, because it is far from clear how to verify the condition (5.91) in most situations of practical importance. In the next subsection we will consider this condition for a simple case: linear advection with first-order upwind spatial discretization. Of course, this is not the spatial discretization one would like to use with a high-order time stepping scheme, but it will give a heuristic explanation for the temporal orders observed in the accuracy experiment in Section 5.4.1.

Remark 5.5.2 The above expressions for the local errors are similar to those given in [24] for implicit-explicit Runge-Kutta methods, and in [40, 41] for a class of implicit additive Runge-Kutta methods with domain decomposition. Apart from the fact that these latter methods are implicit, because they are intended for parabolic problems, an interesting feature is that the matrices I_k are constructed from smooth grid functions, instead of the the step functions (zero-one entries) in this chapter. This can have a positive influence on the accuracy of the schemes. □

Verification of condition (5.91) for linear advection

To study condition (5.91), let us consider linear problems with constant coefficients,

$$u'(t) = Au(t) + g(t)\,. \tag{5.92}$$

Denote $Z = \Delta t A$, $\boldsymbol{Z} = I \otimes Z$ with $I = I_{s\times s}$ the $s \times s$ identity matrix, and

$$\boldsymbol{r}(Z)^T = [r_1(Z), \dots, r_s(Z)] = \Big(\sum_{k=1}^{r} \boldsymbol{b}_k^T \boldsymbol{I}_k \boldsymbol{Z}\Big)\Big(\boldsymbol{I} - \sum_{k=1}^{r} \boldsymbol{A}_k \boldsymbol{I}_k \boldsymbol{Z}\Big)^{-1}. \tag{5.93}$$

In this case we have $\boldsymbol{b}_k^T \boldsymbol{I}_k \boldsymbol{Z} = b_k^T \otimes I_k Z$ and $\boldsymbol{A}_k \boldsymbol{I}_k \boldsymbol{Z} = A_k \otimes I_k Z$. The matrices A_k are strictly lower triangular $s \times s$ matrices, and consequently a product of s such matrices vanishes. Writing the matrix inverse in (5.93) as a power series, it follows that

$$\boldsymbol{r}(Z)^T \boldsymbol{e} = \sum_{l=0}^{s-1} \sum_{k,j_1,\dots,j_l=1}^{r} \big(b_k^T A_{j_1} \cdots A_{j_l} e\big)\, I_k Z\, I_{j_1} Z \cdots I_{j_l} Z\,. \tag{5.94}$$

In the same way it is seen that

$$\begin{aligned} &\boldsymbol{r}(Z)^T \sum_{i=1}^{r} \big(\boldsymbol{c}^q - q\boldsymbol{A}_i \boldsymbol{c}^{q-1}\big) I_i \\ &= \sum_{l=0}^{s-1} \sum_{k,j_1,\dots,j_l,i=1}^{r} \big(b_k^T A_{j_1} \cdots A_{j_l} (c^q - q A_i c^{q-1})\big)\, I_k Z\, I_{j_1} Z \cdots I_{j_l} Z\, I_i\,, \end{aligned} \tag{5.95}$$

If there is a matrix $W \in \mathbb{R}^{m\times m}$ such that $\|W\|_\infty = \mathcal{O}(1)$ and

$$\big(\boldsymbol{r}(Z)^T \boldsymbol{e}\big) W = \boldsymbol{r}(Z)^T \sum_{i=1}^{r} \big(\boldsymbol{c}^q - q\boldsymbol{A}_i \boldsymbol{c}^{q-1}\big) I_i\,, \tag{5.96}$$

then we can take $\xi_n = \frac{1}{q!}\Delta t^q W u^{(q)}(t_n)$ in (5.91). Recall that $\|W\|_\infty = \mathcal{O}(1)$ means that W can be bounded uniformly in the mesh width and dimension m.

Consider as a simple example, the semi-discrete system (5.2) in $\mathbb{R}^m$ with $u_0(t) = 0$, corresponding to first-order upwind discretization of the advection equation with homogeneous inflow condition $u(0,t) = 0$. We take a partitioning $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2 = \{1, 2, \dots, m\}$ with $\mathcal{I}_2 = \{j : \frac{1}{4}m < j \le \frac{3}{4}m\}$, and mesh widths $\Delta x_j = h$ if $j \in \mathcal{I}_1$, $\Delta x_j = \frac{1}{2}h$ if $j \in \mathcal{I}_2$, with $h = 4/(3m)$. In Figure 5.8 we have plotted the norm $\|W\|_\infty$ as function of $m = 20, 40, \dots, 640$ for various values of $\nu = \Delta t/h$ for the schemes TW2 and CS2; the results for SHV2 were similar to those of TW2. In this example, the matrix $\boldsymbol{r}(Z)^T \boldsymbol{e}$ is nonsingular, and it is well-conditioned for $\nu \le 1$. We see that $\|W\|_\infty = \mathcal{O}(1)$ provided that $\nu < 1$, whereas $\|W\|_\infty \sim m$ if $\nu = 1$. Other partitionings $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$ produced similar results.

It is obvious that verification of condition (5.91) would be desirable for nonlinear problems and higher-order (nonlinear) spatial discretizations. Nevertheless, the combination of Proposition 5.5.1 and these experimental bounds for

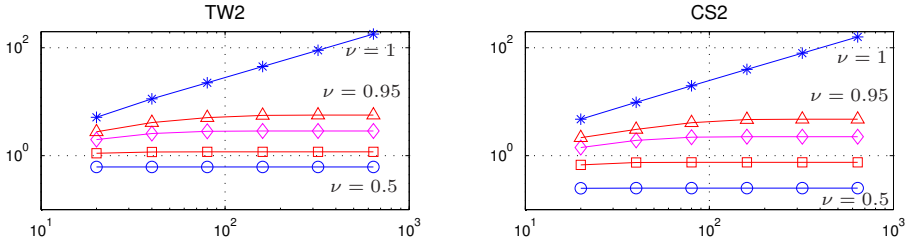


FIGURE 5.8: Norm $\|W\|_\infty$ versus $m = 20, 40, \ldots, 640$ for various values of $\nu = \Delta t/h$ with the schemes TW2 (left) and CS2 (right). Markers: ○ for $\nu = 0.5$, □ for $\nu = 0.75$, ◇ for $\nu = 0.9$, △ for $\nu = 0.95$ and ∗ for $\nu = 1$.

first-order advection discretization does provide a heuristic explanation for the numerical observations in Section 5.4.1 for the advection problem with smooth solution and WENO5 spatial discretization, where we saw convergence of the schemes TW2 and SHV2 with order two in the maximum-norm, and with order one for the CS2 scheme.

5.6 Final remarks

5.6.1 Partitioning based on fluxes

For conservation laws $u_t + f(u)_x = 0$, the semi-discrete system (5.8) will in general be of the form

$$u_j'(t) = F_j(u(t)) = \frac{1}{\Delta x_j}\left(f_{j-\frac{1}{2}}(u(t)) - f_{j+\frac{1}{2}}(u(t))\right), \qquad j \in \mathcal{I} = \{1, 2, \ldots, m\}.$$

Multirate methods can be based on these numerical fluxes $f_{j\pm 1/2}(u)$ rather than in terms of the components $F_j(u)$, and this is not well covered by the above formulations.

Suppose, as an example, that $\mathcal{I}_1 = \{j : j < i\}$ and $\mathcal{I}_2 = \{j : j \geq i\}$. Instead of $F = I_1 F + I_2 F$, we can consider the decomposition $F = F^1 + F^2$ with vector functions F^1 and F^2 whose jth component is given by

$$\left.\begin{aligned} F_j^1(v) &= \frac{1}{\Delta x_j}\left(f_{j-\frac{1}{2}}(v) - f_{j+\frac{1}{2}}(v)\right), & F_j^2(v) &= 0 & &\text{for } j < i, \\ F_j^1(v) &= \frac{1}{\Delta x_i} f_{i-\frac{1}{2}}(v), & F_j^2(v) &= \frac{-1}{\Delta x_i} f_{i+\frac{1}{2}}(v) & &\text{for } j = i, \\ F_j^2(v) &= \frac{1}{\Delta x_j}\left(f_{j-\frac{1}{2}}(v) - f_{j+\frac{1}{2}}(v)\right), & F_j^1(v) &= 0, & &\text{for } j > i. \end{aligned}\right\} \tag{5.97}$$

We can consider any of the above schemes with $I_k F(v)$ replaced by $F^k(v)$. Since we are then dealing with fluxes, mass-conservation is guaranteed at any stage. However, there are two reasons why such schemes were not considered in this chapter.

First, monotonicity assumptions such as (5.13) will not be valid in the maximum-norm with this decomposition. This can be seen already quite easily for the first-order upwind advection discretization (5.2). Writing this system as $u'(t) = Au(t)$, the above decomposition would correspond to $A = AI_1 + AI_2$, that is, $F^k = AI_k$, but it is easy to show that $\|I + \tau AI_k\|_\infty$ is larger than one for any $\tau > 0$.

Secondly, such a decomposition of F can easily lead to inconsistencies, since we do not have $F^k(u(t)) = \mathcal{O}(1)$, no matter how smooth the solution is. For example, for the first-order upwind system (5.2), formula (5.10) with F^k replacing $I_k F$, $k = 1, 2$, leads to method (5.3) rather than (5.4). Using these F^1 and F^2 in (5.9) gives a completely inconsistent result.

5.6.2 Summary and conclusions

In this chapter some multirate schemes based on the forward Euler method and the two-stage explicit trapezoidal rule have been analyzed. All these methods can be written as partitioned Runge-Kutta methods.

For the analysis of the monotonicity properties of the schemes we followed the TVD/SSP framework of [15, 49], assuming monotonicity of one forward Euler step with suitable local time steps. Different monotonicity thresholds were found for maximum-norm monotonicity and maximum principles on the one hand, and the TVD property on the other hand. However, these theoretical differences did not reveal themselves in the numerical tests. In practical situations, the threshold C found for maximum-norm monotonicity seems the most relevant.

Many multirate schemes are not internally consistent. This may lead to low accuracy at interface points. An analysis of the local discretization errors even suggests lack of convergence, but this is too pessimistic. Also for the other schemes, that are internally consistent, propagation of the leading local error terms has to be studied to understand the proper convergence behavior.

Lack of mass conservation seems in many cases not a very serious defect because it only arises at interface points, so it will mainly be felt when a shock or very steep solution gradient passes such an interface. This conclusion is similar as in [54]. Of course, if mass conservation can be built in a scheme without affecting other essential properties, such as internal consistency and computational work per step, this is advisable. For the schemes considered in this chapter lacking mass conservation we did not find such suitable modifications.

The use of a high-order Runge-Kutta methods as basis for a multirate scheme or a partitioned scheme will not directly lead to a high order of accuracy at interface points. The discretization errors have to be considered within the PDE context, leading to expressions for the local errors of the form (5.89) or (5.90). Regarding the semi-discrete as a fixed (non-stiff) ODE will in general lead to a too optimistic estimate of the rate of convergence.

5.7 Appendix: a spatial discretization with TVD limiter on non-uniform grids

As an example of a discretization with limiting we will consider formulas on non-uniform grids that generalize the third-order upwind-biased scheme with the so-called Koren limiter on uniform grids.

5.7.1 Discretization and limiting

For a non-uniform grid with cells $\mathcal{C}_j = (x_j - \frac{1}{2}\Delta x_j, x_j + \frac{1}{2}\Delta x_j)$ and cell-average values u_j, the third-order upwind-biased spatial discretization can be derived by piecewise cubic reconstruction of the primitive grid-function $U_i = \sum_{j \le i} \Delta x_j u_j$ and differentiation.

On $\mathcal{C}_j$ we take $U(x)$ to be the cubic polynomial that passes through the points $(x_{j+k/2}, U_{j+k/2})$, $k = -3, -1, 1, 3$. Then the resulting values

$$u^R_{j-\frac{1}{2}} = U'(x_{j-\frac{1}{2}})\,, \qquad u^L_{j+\frac{1}{2}} = U'(x_{j+\frac{1}{2}})\,,$$

can be used as cell-boundary values in a numerical flux-function. In the following we only give the formulas for the left states $u^L_{j+1/2}$; those for $u^R_{j-1/2}$ are essentially the same, just the mirror image.

By some calculations (with Newton divided differences) it follows that

$$u^L_{j+\frac{1}{2}} = \gamma^L_{-1,j} u_{j-1} + \gamma^L_{0,j} u_j + \gamma^L_{1,j} u_{j+1}\,, \tag{5.98}$$

with coefficients $\gamma^L_{0,j} = 1 - \gamma^L_{-1,j} - \gamma^L_{1,j}$ and

$$\gamma^L_{-1,j} = \frac{-\Delta x_j \Delta x_{j+1}}{(\Delta x_{j-1} + \Delta x_j)(\Delta x_{j-1} + \Delta x_j + \Delta x_{j+1})}\,,$$
$$\gamma^L_{1,j} = \frac{(\Delta x_{j-1} + \Delta x_j)\Delta x_j}{(\Delta x_j + \Delta x_{j+1})(\Delta x_{j-1} + \Delta x_j + \Delta x_{j+1})}\,.$$

This provides the non-limited value.

To apply a limiter, we first write (5.98) in the form

$$u^L_{j+\frac{1}{2}} = u_j + \psi^*_j (u_{j+1} - u_j)\,, \qquad \psi^*_j = \frac{u^L_{j+\frac{1}{2}} - u_j}{u_{j+1} - u_j}\,. \tag{5.99}$$

Next we apply a limiter to this ψ^*_j,

$$\psi_j = \max\big(0\,,\, \min\big(1\,,\, \psi^*_j\,,\, \theta_j\big)\big)\,, \qquad \theta_j = \frac{u_j - u_{j-1}}{u_{j+1} - u_j}\,, \tag{5.100}$$

to obtain the limited value

$$u^L_{j+\frac{1}{2}} = u_j + \psi_j (u_{j+1} - u_j)\,. \tag{5.101}$$

This kind of limiting is often called 'target limitering' because the limited values are taken as close as possible to a target scheme (which is in our case the non-limited scheme) within the monotonicity constraints. It can be applied to any scheme producing non-limited values $u^L_{j+1/2}$. From (5.98), (5.99) it is seen that $\psi^*_j = \gamma^L_{1,j} - \gamma^L_{-1,j}\theta_j$, and therefore the limiter can also be written as

$$\psi_j = \max\big(0\,,\,\min\big(1\,,\,\gamma^L_{1,j} - \gamma^L_{-1,j}\theta_j\,,\,\theta_j\big)\big)\,. \tag{5.102}$$

To see that (5.101) will indeed introduce a spatial discretization with certain monotonicity properties, such as positivity and TVD, note that

$$u^L_{j-\frac{1}{2}} - u^L_{j+\frac{1}{2}} = \rho_j(u_{j-1} - u_j)\,, \qquad \rho_j = 1 - \psi_{j-1} + \psi_j\,/\,\theta_j\,.$$

In view of (5.100) we have $0 \le \psi_{j-1} \le 1$ and $0 \le \psi_j/\theta_j \le 1$, and therefore

$$0 \le \rho_j \le 2\,.$$

As explained in Example 5.2.2, this guarantees max-norm monotonicity and the TVD property for $u_t + f(u)_x = 0$ with $f'(u) \ge 0$ (for the relevant range of u values).

As mentioned already above, the formulas for the right states $u^R_{j-1/2}$ are essentially the same (reflexion around $x_{j-1/2}$), and these will be used if we have $f'(u) < 0$ for all (relevant) u values. With an arbitrary flux function $f(u)$ a suitable flux splitting is to be used, for example the simple Lax-Friedrich splitting given in [33, 48].

Remark 5.7.1 The numerical fluxes $f_{j+1/2}(u) = f(u_{j+1/2})$ of the limited discretization are Lipschitz continuous,

$$|f_{j+1/2}(\tilde u) - f_{j+1/2}(u)| \;\le\; L\|\tilde u - u\|\|_\infty$$

for all $\tilde u = [\tilde u_j]$, $u = [u_j] \in \mathbb{R}^m$. This is not obvious from (5.100), (5.102), because the ratios θ_j will not satisfy a Lipschitz condition. However, if we denote $\sigma_j = u_{j+1} - u_j$, then by considering the different sign possibilities it is seen that

$$u^L_{j+\frac{1}{2}} = u_j + \mathrm{sign}(\sigma_j)\min\big(|\sigma_j|\,,\,\gamma^L_{1,j}|\sigma_j| - \gamma^L_{-1,j}|\sigma_{j-1}|\,,\,|\sigma_{j-1}|\big)$$

if $\mathrm{sign}(\sigma_j) = \mathrm{sign}(\sigma_{j-1})$, and $u^L_{j+1/2} = u_j$ otherwise. From this the Lipschitz condition can be deduced, with Lipschitz constant L determined by the actual grid. □

5.7.2 Accuracy test

Consider the advection equation $u_t + u_x = 0$, $0 < x, t < 1$, with spatial periodicity and initial value $u(x,0) = \sin^4(\pi x)$. The relative L_1-errors of the spatial discretization are given in Table 5.3 for various grids with m points,

$m = 20, 40, 80, 160$. These results are to be compared with those in Appendix B of [5]. The random grids are chosen by first generating random numbers $\sigma_j \in [\frac{1}{2}, 1]$ and then setting $\Delta x_j = \sigma_j / \sum_{k=1}^{m} \sigma_k$. The grids indicated by 'Block1' and 'Block2' are cyclic repetitions of $(\Delta x_1, \Delta x_2, \Delta x_3, \Delta x_4) = (h, 2h, 3h, 4h)$ and $(\Delta x_1, \Delta x_2, \Delta x_3, \Delta x_4) = (h, 2h, 10h, 11h)$, respectively, with appropriate $h = 4/(10m)$, $h = 4/(14m)$, respectively.

TABLE 5.3: Relative L_1-errors for scalar advection on non-uniform grids

	Uniform	Random	Block 1	Block 2
Non-lim., $m = 20$	$4.79 \cdot 10^{-2}$	$5.14 \cdot 10^{-2}$	$6.06 \cdot 10^{-2}$	$9.65 \cdot 10^{-2}$
Non-lim., $m = 40$	$6.82 \cdot 10^{-3}$	$7.49 \cdot 10^{-3}$	$9.13 \cdot 10^{-3}$	$1.58 \cdot 10^{-2}$
Non-lim., $m = 80$	$8.70 \cdot 10^{-4}$	$9.49 \cdot 10^{-4}$	$1.18 \cdot 10^{-3}$	$2.05 \cdot 10^{-3}$
Non-lim., $m = 160$	$1.09 \cdot 10^{-4}$	$1.19 \cdot 10^{-4}$	$1.49 \cdot 10^{-4}$	$2.60 \cdot 10^{-4}$
Limited, $m = 20$	$6.57 \cdot 10^{-2}$	$6.79 \cdot 10^{-2}$	$9.35 \cdot 10^{-2}$	$1.45 \cdot 10^{-1}$
Limited, $m = 40$	$1.36 \cdot 10^{-2}$	$1.49 \cdot 10^{-2}$	$2.02 \cdot 10^{-2}$	$3.32 \cdot 10^{-2}$
Limited, $m = 80$	$2.65 \cdot 10^{-3}$	$2.97 \cdot 10^{-3}$	$4.25 \cdot 10^{-3}$	$7.56 \cdot 10^{-3}$
Limited, $m = 160$	$4.97 \cdot 10^{-4}$	$5.73 \cdot 10^{-4}$	$8.11 \cdot 10^{-4}$	$1.58 \cdot 10^{-3}$

The results compare favourably to those in [5], where it should be noted that the random grid used here has more variation in [5] and also the initial profile has been slightly changed to make it periodic.

We also note that the above limiter does not fit into the framework of slope limiting with linear reconstruction considered in [5]. There it is required that on each cell $\mathcal{C}_j$ we have an approximation $u(x) = u_j + (x - x_j)s_j$, with slope s_j that may be limited, and then

$$u^R_{j-\frac{1}{2}} = u_j - \frac{1}{2}\Delta x_j s_j\,, \qquad u^L_{j+\frac{1}{2}} = u_j + \frac{1}{2}\Delta x_j s_j\,.$$

To achieve this in the above algebraic framework one needs a certain 'symmetry' condition to ensure that u_j is the average of $u^R_{j-1/2}$ and $u^L_{j+1/2}$.

The spatial discretization used in [8] is of the same form as (5.102) but with different coefficients $\gamma_{k,j}$. In the above accuracy test this scheme gave less accurate results, due to the fact that then the non-limited scheme is only of order two. The errors with limiter were then a factor three to four larger than in Table 5.3 on the fine grids, $m = 160$.

Finally we note that the limited schemes used in [54] are based on scaled ratios $\theta_j = \sigma_{j-1}/\sigma_j$ with $\sigma_k = (u_{k+1} - u_k)/\Delta x_k$. It is not too difficult to show that such schemes are not TVD or positivity preserving, but in tests they do perform quite well; there are overshoots, but these are very minor. Nevertheless, to remain within the theoretical framework outlined in Section **??**, the discretization (5.102) seems preferable.

Summary

For large systems of ordinary differential equations (ODEs), some components may show a more active behavior than others. To solve such problems numerically, multirate integration methods can be very efficient. These methods enable the use of large time steps for slowly varying components and small steps for rapidly varying ones. In this thesis we design, analyze and test multirate methods for the numerical solution of ODEs.

A self-adjusting multirate time stepping strategy is presented in Chapter 1. In this strategy the step size for a particular system component is determined by the local temporal variation of this solution component, in contrast to the use of a single step size for the whole set of components as in the traditional methods. The partitioning into different levels of slow to fast components is performed automatically during the time integration. The number of activity levels, as well as the component partitioning, can change in time. Numerical experiments confirm that with our strategy the efficiency of time integration methods can be significantly improved by using large time steps for inactive components, without sacrificing accuracy.

A multirate scheme, consisting of the θ-method with one level of temporal local refinement, is analysed in Chapter 2. Missing component values, required during the refinement step, are computed using linear or quadratic interpolation. This interpolation turns out to be important for the stability of the multirate scheme. Moreover, the analysis shows that the use of linear interpolation can lead to an order reduction for stiff systems. The theoretical results are confirmed in numerical experiments.

Two multirate strategies, *recursive refinement* and the *compound step* strategy are compared in Chapter 3. The recursive refinement strategy has somewhat larger asymptotic stability regions than the compound step strategy. The compound step strategy, by avoiding the extra work of doing the macro step for all the components, looses some stability properties compared to the recursive refinement strategy. It can also lead to more complex algebraic implicit systems, which are difficult to solve numerically.

The construction of higher-order multirate Rosenbrock methods is discussed in Chapter 4. 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 recommend a strategy of avoidance of the order reduction for problems with a stiff source term. A multirate method based on the fourth-order Rosenbrock method RODAS and its third-order dense output has been designed. This multirate RODAS method has shown very good results in numerical experiments, and it is clearly more efficient than other

considered multirate methods in these tests.

Explicit multirate and partitioned Runge-Kutta schemes for semi-discrete hyperbolic conservation laws are analysed in Chapter 5. It appears that, for the considered class of multirate methods, it is not possible to construct a multirate scheme which is both locally consistent and mass-conservative. The analysis shows that, in spite of local inconsistencies, global convergence is still possible in all grid points.

Samenvatting

Voor grote systemen van gewone differentiaalvergelijkingen kunnen sommige componenten een actiever gedrag vertonen dan andere. Om zulke problemen numeriek op te lossen kunnen zogenaamde multirate methoden zeer efficiënt zijn. Bij zulke methoden is het mogelijk om een grote tijdstap te nemen voor langzaam variërende componenten en kleine tijdstappen voor componenten met een snelle variatie. In dit proefschrift komen ontwerp, analyse en experimentele resultaten aan de orde van multirate methoden voor het numeriek oplossen van gewone differentiaal vergelijkingen.

Een zelf-regulerende multirate strategie wordt gepresenteerd in hoofdstuk 1. Bij deze strategie is de stapgrootte voor een zekere component van het systeem bepaald door de lokale verandering in tijd van deze component. Dit is anders bij traditionele methoden waar één en dezelfde tijdstap gebruikt wordt voor alle componenten. De partitionering in verschillende niveaus van activiteit, van snelle tot langzame componenten, wordt automatisch uitgevoerd tijdens de tijdsintegratie. Het aantal activiteiten-niveaus, alsmede de componenten partitionering, kan variëren in tijd. Numerieke experimenten bevestigen dat de efficiëntie van tijdsintegratie-methoden met onze strategie aanzienlijk verbeterd kan worden door grote tijdstappen te gebruiken voor de inactieve componenten, zonder aantasting van de nauwkeurigheid.

Een multirate schema, bestaande uit de zogenaamde θ-methode met één niveau van lokale verfijning in tijd, wordt geanalyseerd in hoofdstuk 2. Niet-aanwezige waarden van componenten die vereist worden tijdens de verfijningsstap worden berekend met lineaire of quadratische interpolatie. De keuze van interpolatie blijkt zeer belangrijk voor de stabiliteit van het multirate schema. De analyse laat bovendien zien dat het gebruik van lineaire interpolatie kan leiden tot een reductie van de orde van nauwkeurigheid van het schema voor stijve problemen.

Twee multirate strategieën, *recursieve verfijning* en de *compound step strategie*, worden vergeleken in hoofdstuk 3. De recursieve verfijnings strategie heeft ietwat grotere gebieden van asymptotische stabiliteit. De compound step strategie vermijdt het extra werk van de macro-stap voor alle componenenten, maar dit leidt tot verlies van zekere stabiliteits eigenschappen in vergelijking met de recursieve verfijnings strategie. Bovendien geeft de compound step strategie aanleiding tot complexere algebraïsche impliciete systemen, die moeilijk numeriek op te lossen zijn.

De constructie van hogere-orde multirate Rosenbrock methoden wordt besproken in hoofdstuk 4. Een onjuiste behandeling van stijve brontermen en het gebruik van lage-orde interpolanten kan leiden tot orde-reductie, waarbij we een

lagere orde van consistentie krijgen dan voor niet-stijve problemen. Een aanpak wordt aanbevolen waarmee deze orde-reductie voor problemen met een stijve bronterm vermeden wordt. Een multirate methode gebaseerd op de vierde-orde Rosenbrock methode RODAS met een derde-orde dense-output formule is ontworpen. Deze multirate RODAS methode heeft zeer goede resultaten opgeleverd in numerieke experimenten, en is duidelijk efficiënter dan andere multirate methoden in deze experimenten.

Expliciete multirate methoden en gepartitioneerde Runge-Kutta methoden, voor semi-discrete hyperbolische behoudswetten, worden geanalyseerd in hoofdstuk 5. Het blijkt dat het voor de beschouwde klasse van multirate methoden niet mogelijk is om lokale consistentie te combineren met massabehoud. De analyse geeft aan dat ondanks lokale inconsistenties toch globale convergentie verkregen kan worden in alle roosterpunten.

Acknowledgements

This thesis records the research that I conducted between 2004 and 2008, as a member of the cluster Modelling, Analysis and Simulation of the Centrum voor Wiskunde en Informatica (CWI) in Amsterdam.

First of all, I would like to thank my daily supervisor Willem Hundsdorfer and my promotor Jan Verwer for the many things I learned from them, for sharing great ideas and for their valuable guidance. It was a real pleasure to work with Willem and Jan, and I am very grateful they gave me the opportunity to work on this project.

I would like to acknowledge the Stichting Van Beuningen/Peterich Fonds and Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) for funding this project.

Next, I would like to thank Lennart Edsberg and Numerisk Analys och Datalogi department (NADA) for the wonderful Master's programme in Scientific Computing. It was a fascinating introduction to the field and it helped me to enrich my experience.

I am also grateful to Arie Verhoeven and Jan ter Maten for the interesting discussions we had, and for the test problems from the industry.

I would like to thank all my CWI colleagues for creating a nice and friendly working environment. I enjoyed all lunches, coffee breaks and discussions we had. In particular, my office mates: Ismail, Nga, Yousaf and Sveta; Anna, Maksat and Chao. Thanks to Jan Schipper and Jos van der Werf for their help on printing the thesis; Nada Mitrovic and Susanne van Dam for their help with all kind of documents.

Furthermore, thanks to my "Swedish" and "Dutch" friends with whom I shared all the good and bad times, dinners, parties, bike trips,... To Anna & Vasiliy, Carolina, Corina, Dmitry, Eugen, Ira, Janis, Julieta & Victor, Lilit, Lili & Aurelian, Loredana & Balder, Nga & Thieu, Oxana, Stephanie & Nils, Sveta & Denis, Tamara & Denis and Yousaf.

My special regards to Dmitry and Eugen, my very first friends in Sweden, for all their support and help, for the endless discussions about the meaning of life, and of course for *palinca* :) Nga and Thieu, thanks for the interesting discussions about mathematics and life, and for the numerous exceptional Vietnamese dinners. Carolina, thanks for the continuous keep-alive MSN messages and for the unbeatable cakes. Loredana, thanks for your care and for the tasty dinners. Lilit, thanks for the friendship and for the unforgettable huge tray of

home made Armenian *pakhlava*. Anna, thanks for the help with the cover of the thesis. Vlad, my best friend in Moldova for many years now, thanks for your friendship, support and encouragement from so far away. Thanks to all of you. I know that I always can count on you.

I am very grateful to my dear Alina for the idea of the cover design, for her inexhaustible support and understanding, and continued encouragement over the last years.

Finally, I would like to thank my sister Ira and her husband Jörg for their infinite support throughout everything. My special thanks to my father, who influenced my personality more than anybody else, making me able to go where I am now. I would like to express my gratitude and love to my mother, who always believed in me and who was helping me in more ways than I can count. Mother, I will always be proud of you.

Bibliography

[1] I. Alonso-Mallo, B. Cano, *Spectral/Rosenbrock discretizations without order reduction for linear parabolic problems*. Appl. Numer. Math. 41 (2002), 247–268.

[2] A. Bartel, *Generalised Multirate. Two ROW-type versions for circuit simulation*. Unclass. Natlab Report No. 2000/84, Philips Electronics, 2000.

[3] A. Bartel, M. Günther, *A multirate W-method for electrical networks in state space formulation*. J. Comp. Appl. Math. 147 (2002), 411–425.

[4] A. Bellen, M. Zennaro, *Stability properties of interpolants for Runge-Kutta methods*. SIAM J. Numer. Anal. 25 (1988), 411–432.

[5] M. Berger, M.J. Aftosmis, S.M. Murman, *Analysis of slope limiters on irregular grids*. AIAA Paper 2005-0490, 2005.

[6] K. Burrage, J.C. Butcher, *Nonlinear stability of a general class of differential equation methods*. BIT 20 (1980), 185–203.

[7] J.C. Butcher, *Numerical Methods for Ordinary Differential Equations*. Second edition, Wiley, 2003.

[8] E.M. Constantinescu, A. Sandu, *Multirate timestepping methods for hyperbolic conservation laws*. Report TR-06-15 (913), Dept. Comp. Sc. Virginia Tech, 2006.

[9] C. Dawson, R. Kirby, *High resolution schemes for conservation laws with locally varying time steps*. SIAM J. Sci. Comput. 22 (2000), 2256–2281.

[10] C. Engstler, C. Lubich, *Multirate extrapolation methods for differential equations with different time scales*. Computing 58 (1997), 173–185.

[11] C. Engstler, C. Lubich, *MUR8: A multirate extension of the eight-order Dormand-Prince method*. Appl. Numer. Math. 25 (1997), 185–192.

[12] D. Estep, M.G. Larson, R.D. Williams, *Estimating the Error of Numerical Solutions of Systems of Reaction-Diffusion Equations*. Mems. Amer. Math. Soc. 146 (2000), no. 696.

[13] I. Faragó, C. Palencia, *Sharpening the estimate of the stability constant in the maximum-norm of the Crank-Nicolson scheme for the one-dimensional heat equation*. Appl. Numer. Math. 42 (2002), 133–140.

[14] C. Gear, D. Wells, *Multirate linear multistep methods*. BIT 24 (1984), 484–502.

[15] S. Gottlieb, C.W. Shu, E. Tadmor, *Strong stability preserving high-order time discretization methods*. SIAM Review 42 (2001), 89–112.

[16] M. Günther, A. Kværnø, P. Rentrop, *Multirate partitioned Runge-Kutta methods*. BIT 41 (2001), 504–514.

[17] L. Ferracina, M.N. Spijker, *An extension and analysis of the Shu-Osher representation of Runge-Kutta methods*. Math. Comp. 74 (2005), 201–219.

[18] E. Hairer, S.P. Nørsett, G. Wanner, *Solving Ordinary Differential Equations I – Nonstiff Problems*. Second edition, Springer Series Comput. Math. 8, Springer, 1993.

[19] E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II – Stiff and Differential-Algebraic Problems*. Second edition, Springer Series in Comp. Math. 14, Springer, 1996.

[20] A. Harten, *High resolution schemes for hyperbolic conservation laws*. J. Comput. Phys. 49 (1983), 357–393.

[21] I. Higueras, *Representations of Runge-Kutta methods and strong stability preserving methods*. SIAM J. Numer. Anal. 43 (2005), 924–948.

[22] I. Higueras, *Strong stability for additive Runge-Kutta methods*. SIAM J. Numer. Anal. 44 (2006), 1735–1758.

[23] R.A. Horn, C.R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[24] W. Hundsdorfer, S.J. Ruuth, *IMEX extensions of linear multistep methods with general monotonicity and boundedness properties*. CWI Report MAS-E0621, Amsterdam, 2006. To appear in J. Comput. Phys.

[25] W. Hundsdorfer, V. Savcenco, *Analysis of a multirate theta-method for stiff ODEs*. To appear in J. Appl. Num. Math.

[26] W. Hundsdorfer, A. Mozartova, V. Savcenco, *Analysis of explicit multirate and partitioned Runge-Kutta schemes for conservation laws*. CWI report, MAS-E0715, 2007.

[27] W. Hundsdorfer, J.G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer Series in Comp. Math. 33, Springer, 2003.

[28] J. Jansson, A. Logg, *Algorithms for multi-adaptive time stepping*. Chalmers Finite Element Center, Preprint 2004-13, 2004.

[29] P. Kaps, P. Rentrop, *Generalized Runge-Kutta Methods of order four with stepsize control for stiff ordinary differential equations*. Numer. Math. 33 (1979), 55–68.

[30] R. Kirby, *On the convergence of of high resolution methods with multiple time scales for hyperbolic conservation laws*. Math. Comp. 72 (2003), 1239–1250.

[31] A. Kværnø, *Stability of multirate Runge-Kutta schemes*. Int. J. Differ. Equ. Appl. 1A (2000), 97–105.

[32] J.D. Lambert, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. Wiley, 1991.

[33] R.J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Appl. Math., Cambridge Univ. Press, 2002.

[34] A. Logg, *Multi-adaptive Galerkin methods for ODEs I*. SIAM J. Sci. Comput. 24 (2003), 1879–1902.

[35] A. Logg, *Multi-adaptive Galerkin methods for ODEs II. Implementation and applications*. SIAM J. Sci. Comput. 25 (2003), 1119–1141.

[36] N.M. Maurits, H. van der Ven, A.E.P. Veldman, *Explicit multi-time stepping methods for convection dominated flow problems*. Comput. Meth. Appl. Mech. Engrg. 157 (1998), 133–150.

[37] S. Osher, R. Sanders, *Numerical approximations to nonlinear conservation laws with locally varying time and space grids*. Math. Comp. 41 (1983), 321–336.

[38] A. Ostermann, M. Roche, *Rosenbrock methods for partial differential equations and fractional orders of convergence*. SIAM J. Numer. Anal. 30 (1993), 1084–1098.

[39] A. Ostermann, *Continuous extensions of Rosenbrock-Type methods*. Computing 44 (1990), 59–68.

[40] L. Portero, B. Bujanda, J.C. Jorge, *A combined fractional step domain decomposition method for the numerical integration of parabolic problems*, Lect. Notes in Comp. Sc. 3019 (2004), 1034–1041.

[41] L. Portero, *Fractional step Runge-Kutta methods for multidimensional evolutionary problems with time-dependent coefficients and boundary conditions*. Thesis, Univ. Publ. Navarra, Pamplona, 2007.

[42] R.D. Richtmyer, K.W. Morton, *Difference Methods for Initial-Value Problems*. Second edition, John Wiley & Sons, Interscience Publishers, 1967.

[43] G. Rodriguez-Gómez, P. González-Casanova, J. Martinez-Carballido, - *Computing general companion matrices and stability regions of multirate methods,* Int. J. Numer. Methods Engrg. 61 (2004), 255–273.

[44] J. Sand, S. Skelboe, *Stability of backward Euler multirate methods and convergence of waveform relaxation.* BIT 32 (1992), 350–366.

[45] V. Savcenco, *Comparison of the asymptotic stability properties for two multirate strategies.* to appear in Journal Comp. Appl. Math.

[46] V. Savcenco, *Construction of high-order multirate Rosenbrock methods for stiff ODEs.* CWI report, MAS-E0716, 2007.

[47] V. Savcenco, W. Hundsdorfer, J.G. Verwer, *A multirate time stepping strategy for stiff ordinary differential equations.* BIT 47 (2007), 137–155.

[48] C.W. Shu, *High order ENO and WENO schemes for computational fluid dynamics.* In: *High-Order Methods for Computational Physics*, Eds. T.J. Barth, H. Deconinck, Lect. Notes Comp. Sc. Eng. 9, Springer, 1999, 439–582.

[49] C.W. Shu, S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes.* J. Comput. Phys. 77 (1988), 439–471.

[50] S. Skelboe, *Stability properties of backward differentiation multirate formulas.* Appl. Numer. Math. 5 (1989), 151–160.

[51] L.F. Shampine, *Numerical Solution of Ordinary Differential Equations.* Chapman & Hall, 1994.

[52] M.N. Spijker, *Stepsize restrictions for general monotonicity in numerical initial value problems.* SIAM J. Numer. Anal. 45 (2007), 1226–1245.

[53] M. Striebel, M. Günther, *A charge oriented mixed multirate method for a special class of index-1 network equations in chip design.* Appl. Numer. Math. 53 (2005), 489–507.

[54] H.Z. Tang, G. Warnecke, *High resolution schemes for conservation laws and convection-diffusion equations with varying time and space grids.* J. Comput. Math. 24 (2006), 121–140.

[55] A.Verhoeven, A.El Guennouni, E.J.W. ter Maten, R.M.M.Mattheij, *A general compound multirate method for circuit simulation problems.* In: A.M. Anile, G. Ali, G. Mascali, "Scientific Computing in Electrical Engineering", Series Mathematics in Industry, ECMI, Vol. 9, pp. 143–150, Springer, 2006.

[56] A. Verhoeven, B. Tasic, T.G.J. Beelen, E.J.W. ter Maten, R.M.M. Mattheij, *Automatic partitioning for multirate methods.* In: G. Ciuprina, D. Ioan, "Scientific Computing in Electrical Engineering", Series Mathematics in Industry, Vol. 11, pp. 229–236, Springer, 2007.