1991

P.J. van der Houwen, B.P. Sommeijer, K. Strehmel

Smoothed Runge-Kutta methods in the method of lines

Department of Numerical Mathematics Report NM-R9101 January

CWI, nationaal instituut voor onderzoek op het gebied van wiskunde en informatica

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

Smoothed Runge-Kutta Methods in the Method of Lines

P.J. van der Houwen & B.P. Sommeijer
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

K. Strehmel
Martin Luther Universität,
Halle, Deutschland

The effect of introducing smoothing matrices into Runge-Kutta methods for solving semidiscrete partial differential equations is investigated. In particular, the paper focusses on stability and accuracy. It is shown that the stiffness of the initial-value problem can be removed by inserting simply structured smoothing matrices that are designed for use on vector computers so that the increase of the computational costs is rather modest. A number of explicitly given methods together with their smoothing error and stability boundaries are given. Numerical examples illustrate the theoretical results.

1980 Mathematics Subject Classification: 65M10, 65M20; CR: G1.8
Key Words & Phrases: partial differential equations, initial-boundary-value problems, stability, smoothing.

1. INTRODUCTION

Let the semidiscrete partial differential equation be given by the system of ordinary differential equations (ODEs)

$$\frac{dy(t)}{dt} = f(t,y(t)). \tag{1.1}$$

Such differential equations are usually very stiff, so that in most cases *implicit* integration methods are used in order to avoid the rather restrictive stepsize condition associated with *explicit* methods when integrating stiff systems. However, when using implicit methods, one is faced with the problem of solving implicit relations which may require a lot of computational effort, particularly in the case of higher-dimensional problems. This motivated Wubs [9] to introduce *righthand side smoothing*, that is, instead of integrating (1.1), he proposed to integrate the 'smoothed' differential equation

$$\frac{dy(t)}{dt} = Sf(t,y(t)), \tag{1.2}$$

where S is a smoothing matrix which 'removes the stiffness from the the righthand side function', so that explicit methods are now feasible for integrating (1.2). The resulting method will be called a *righthand side smoothing method* or briefly RHS method.

One drawback of RHS methods is the introduction of smoothing errors causing a difference between the solutions of the equations (1.1) and (1.2), and hence, a difference between the numerical solutions of (1.1) and (1.2). Since this difference is larger as the matrix S removes more stiffness from **f**, we pay a high price for relaxing the stability condition. For this reason, it was proposed in [4] to replace S by the smoothing matrix (2I - S)S. This smoothing matrix does reduce the smoothing error, but it also halves the stability boundary of the method.

A second unsatisfactory aspect of RHS methods is that until now a rigorous stability analysis is based on smoothing matrices that are polynomials of the normalized Jacobian of **f**, that is, $S=P(J/\rho(J))$, where $\rho(J)$ denotes the spectral radius of the Jacobian J (cf. [3]).

In this paper, we investigate whether the smoothing error can be reduced by inserting the smoothing matrix S directly into the numerical method, rather than inserting S into the differential equation, with a minimal reduction of the stability boundary. In particular, this will be done for Runge-Kutta methods (RK methods), and the resulting methods will be called *smoothed Runge-Kutta methods* (SRK methods). Secondly, we present a rigorous stability analysis for smoothing matrices S=P(D) where D is not required to be related to the Jacobian matrix J.

Report NM-R9101
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

2. SMOOTHED RUNGE-KUTTA METHODS

Consider the explicit m-stage RK method (see e.g. [1]):

$$\text{(2.1a)} \qquad y_{n+1} = y_n + h \sum_{i=1}^{m} b_i \, \mathbf{f}(t_n + c_i h, \mathbf{Y}_i),$$

where the intermediate vectors $\mathbf{Y}_i$ are defined by

$$\text{(2.1b)} \qquad \mathbf{Y}_i = y_n + h \sum_{j=1}^{i-1} a_{ij} \, \mathbf{f}(t_n + c_j h, \mathbf{Y}_j), \quad i = 1, \dots, m.$$

The stability condition of (2.1) is of the form

$$\text{(2.2)} \qquad h \le \frac{\beta}{\rho(J)},$$

where $\rho(J)$ denotes the spectral radius of the Jacobian matrix $J=\partial\mathbf{f}/\partial y$, and β is a usually small constant (stability boundary) defined by the RK method. If (2.1) is applied to a semidiscrete partial differential equation, then $\rho(J)$ is usually extremely large, so that condition (2.2) prescribes stepsizes that are smaller than accuracy conditions require. In order to relax the stability conditions of explicit RK methods, we replace the parameters a_{ij} and b_i by polynomials A_{ij} and B_i of a matrix S, i.e., $A_{ij}=A_{ij}(S)$, $B_i=B_i(S)$, to obtain the SRK method:

$$\text{(2.3a)} \qquad y_{n+1} = y_n + h \sum_{i=1}^{m} B_i(S) \, \mathbf{f}(t_n + c_i h, \mathbf{Y}_i),$$

$$\text{(2.3b)} \qquad \mathbf{Y}_i = y_n + h \sum_{j=1}^{i-1} A_{ij}(S) \, \mathbf{f}(t_n + c_j h, \mathbf{Y}_j), \quad i = 1, \dots, m,$$

$$\text{(2.3c)} \qquad c_i = \sum_{j=1}^{i-1} A_{ij}(1), \quad i = 1, \dots, m.$$

Methods of the form (2.3) fit into the class of *generalized Runge-Kutta methods* studied in van der Houwen [2], and into the class of *adaptive Runge-Kutta methods* studied in Strehmel [7]. These papers almost exclusively deal with the case where the matrix S is some approximation to the matrix hJ. From a computational point of view, the matrix hJ may be rather expensive, so that we shall look for matrices S that are more effective with respect to the computational effort involved.

In this paper, we shall identify S with a smoothing matrix which causes a damping of the high frequencies occurring in the Fourier expansion of the grid function to which it is applied. Since the usually severe time step restriction associated with the application of explicit methods to semidiscrete partial differential equations stems from the high frequencies occurring in the righthand side function **f**, we expect that removing the high frequencies from **f** stabilizes the method and therefore relaxes the stability condition.

2.1. Smoothing matrices

As observed above, the matrix S will be chosen such that it removes the high frequencies from the Fourier expansion of the grid function to which it is applied. Smoothing matrices have been constructed in, e.g., [5], [6] and [8]. More recently, smoothing matrices based on Chebyshev polynomials were constructed in [9], [3] and [4]. In this paper, we shall use the Chebyshev-polynomials-based smoothing matrices. An outline of their construction is given below.

Let D be a difference matrix with a complete eigensystem $\{\mathbf{e}_j\}$ and eigenvalues μ_j, and let the vector $\mathbf{v}$ to be smoothed be expanded with respect to the eigensystem of D:

$$\text{(2.4)} \qquad \mathbf{v} = \sum_j a_j \mathbf{e}_j.$$

Typically, the frequency of the eigenvectors of D is proportional to the magnitude of the corresponding eigenvalue. In such cases, we can develop smoothing matrices S=P(D) by constructing a polynomial P(z) whose magnitude becomes smaller if eigenvalues of D of larger magnitude are substituted. Since,

$$\text{(2.5)} \qquad P(D)\mathbf{v} = \sum_j P(\mu_j) a_j \mathbf{e}_j,$$

we see that the higher frequencies are increasingly stronger damped.

For efficiency reasons, it is desirable that D is independent of the righthand side function **f**. On the other hand, we experienced that, for accuracy reasons, D should resemble (to some extent) the normalized Jacobian J/ρ(J) of **f**. It turns out that for sufficiently smooth righthand sides **f** originating from one-dimensional Dirichlet problems defined on a uniform grid of width Δ, a suitable difference matrix D is given by

$$(2.6) \qquad D = \frac{1}{4}\begin{pmatrix} 0 & & & & & 0 \\ 1 & -2 & 1 & & & \\ & & \cdot & \cdot & \cdot & \\ & & & 1 & -2 & 1 \\ 0 & & & & & 0 \end{pmatrix}.$$

Notice that this matrix is identical with the normalized Jacobian associated with the standard symmetric semidiscretization of the Dirichlet problem for the equation $u_t=u_{xx}$.

The most simple polynomial P(z) defining the smoothing matrix S=P(D) is given by P(z)=1+z. In the case of (2.6) this leads to the smoothing matrix

$$(2.7) \qquad S = P(D) = \frac{1}{4}\begin{pmatrix} 4 & & & & & 0 \\ 1 & 2 & 1 & & & \\ & & \cdot & \cdot & \cdot & \\ & & & 1 & 2 & 1 \\ 0 & & & & & 4 \end{pmatrix}$$

which is easily recognized as an averaging matrix. In the case of the difference matrix defined in (2.6), the eigenvalues of D are in the interval [-1,0], so that P(D) has eigenvalues in the interval [0,1].

Let us consider more closely smoothing matrices based on difference matrices D with eigenvalues in [-1,0]. There are, of course, many possibilities to define polynomials P(z) which generate smoothing matrices with the required spectral properties. For example, a rather effective smoother is defined by the polynomial (cf. [3])

$$(2.8) \qquad P(z) := \frac{T_k(1+2z) - 1}{2zk^2}, \quad T_k(\cos\theta) := \cos(k\theta).$$

In actual computation, there are various options for implementing the smoothing matrix S=P(D). The most straightforward way consists of computing the elements of the matrix P(D) in advance. This is feasible for a simply structured D in one spatial dimension, but in more dimensions it is not recommendable. However, what one might do in higher-dimensional problems, is the application of the one-dimensional smoothing matrices successively in the spatial directions, or, alternatively, the use of the recursive relations satisfied by Chebyshev polynomials. If k is the degree of the polynomial P(z), then the generation of a smoothed vector **w**=P(D)**v** requires k matrix-vector multiplications with the matrix D. This implementation is simple and applies to any matrix D in any number of spatial dimensions, but can be rather time-consuming when run on a computer. A third possibility is offered by the factorization property of the polynomial (2.8) which allows a nice factorization of the corresponding smoothing matrices:

Theorem 2.1. Let the factors $F_j(z)$ be generated by

$$F_1(z) := 1 + z, \ F_j(z) := [1 - 2F_{j-1}(z)]^2, j \geq 2,$$

and let $k=2^q$, then the polynomial (2.8) can be obtained by the factorization formula

$$P(z) = F_q(z)F_{q-1}(z) \ldots F_1(z). \ \square$$

Thus the matrix P(D) can be obtained as the product of the factor matrices $F_j(D)$. Based on this factorization property, an extremely efficient algorithm can be constructed. Notice that by computing the factor matrices $F_j(D)$ in advance, the generation of the smoothed vector P(D)v requires only ${}^2\log(k)$ matrix-vector multiplications.

The reduced number of matrix-vector multiplications when using an implementation based on Theorem 2.1 is only an advantage if the factor matrices $F_j(D)$ have relatively few nonzero elements in each row, otherwise, it may be cheaper to compute in advance the smoothing matrix S itself. In one-dimensional problems, many difference matrices (for example, the matrix D defined by (2.6)) do lead to simple factor matrices and their precomputation is certainly feasible from a computational point of view. In higher-dimensional problems, matters are different. One possibility is based on the approach of applying one-dimensional matrices in the successive spatial directions. For example, in two-dimensional problems we associate with the two spatial directions the smoothing matrices $S_1=P(D_1)$ and $S_2=P(D_2)$, where P is defined by (2.8) and D_1 and D_2 are of the type (2.6). Then S may be defined by $S=S_1S_2$.

2.2. The stability polynomial

The (linear) stability of SRK methods of the form (2.3) is governed by its stability polynomial. In order to derive this polynomial, the method is applied to the linear test equation y'=Jy, where J is a constant matrix. In this paper, we shall concentrate on the special case where the polynomials $A_{ij}(x)$ and $B_j(x)$ all have a zero at x=0, i.e.

(2.9) $\quad A_{ij}(x) = x\bar{A}_{ij}(x), \ B_i(x) = x\bar{B}_i(x).$

We observe that such SRK methods may also be interpreted as the general SRK method (2.3) applied to the 'smoothed' differential equation (1.2). Furthermore, if $\bar{A}_{ij}$ and $\bar{B}_i$ are constants, then the SRK method is equivalent with a conventional RK method with scalar coefficients applied to (1.2). A related class of righthand-side-smoothing methods is obtained by setting

(2.9') $\quad A_{ij}(x) = (2 - x)x\bar{A}_{ij}, \ B_i(x) = (2 - x)x\bar{B}_i,$

where $\bar{A}_{ij}$ and $\bar{B}_i$ are constants. The corresponding methods are equivalent to applying a conventional RK method with scalar coefficients to the 'smoothed' equation (cf. [4])

(2.10) $\quad \frac{dy(t)}{dt} = (2I - S)Sf(t,y(t)).$

If the SRK method is applied to the test equation y'=Jy and if (2.9) is satisfied, then we obtain the recursion

$$y_{n+1} = R(hSJ,S)\ y_n,$$

where R(hSJ,S) is the *amplification matrix* associated with the SRK method and R is the *stability polynomial* which can be shown to be of the form

(2.11) $\quad R(z,x) := \det\left[I - z\bar{A}(x) + z\mathbf{e}\bar{\mathbf{b}}^T(x)\right].$

Here $\mathbf{e}$ is the m-vector with unit entries, $\bar{A}(x)$ is the matrix with entries $\bar{A}_{ij}(x)$, and $\bar{\mathbf{b}}(x)$ is the vector with entries $\bar{B}_i(x)$.

If we assume that SJ and S share the same eigensystem, then a particular eigenvector in the expansion of y_n is multiplied by the scalar $R(h\lambda(SJ),\lambda(S))$, where $\lambda(SJ)$ and $\lambda(S)$ denote the eigenvalues of SJ and S, respectively. This justifies to define the *stability region* associated with R(z,x) by the set $\mathbb{S}$ of points in the (z,x)-space where R(z,x) is bounded by 1 (compare a similar definition in conventional RK methods). Denoting the spectrum of a matrix M by $\Lambda(M)$, the stability condition becomes

$$h\Lambda(SJ) \times \Lambda(S) \ \text{ in } \mathbb{S}.$$

In this paper, it is always assumed that S has its eigenvalues in the interval [0,1], so that the range of the variable x can be restricted to [0,1]. For example, this is the case if the polynomial P is defined by (2.8) and if the matrix D has its eigenvalues in the interval [-1,0] (e.g., the matrix D defined by (2.6) has this property).

We shall discuss methods for parabolic and hyperbolic equations. In order to determine the parabolic and hyperbolic stability region we define for each fixed x in [0,1] the real (or parabolic) stability boundary $\beta_{real}(x)$ and the imaginary (or hyperbolic) stability boundary $\beta_{imag}(x)$ of the polynomial R(z,x). The parabolic stability region is defined by

$$\mathbb{R}_{par} := \{(z,x): \ -\beta_{real}(x) \le z \le 0, \ 0 \le x \le 1\},$$

and the hyperbolic stability region is defined by

$$\mathbb{R}_{hyp} := \{(z,x): z = iy, -\beta_{imag}(x) \le y \le \beta_{imag}(x), \ 0 \le x \le 1\}.$$

In these cases, the corresponding condition on the stepsize h assumes the form (cf. condition (2.2))

(2.12) $\quad h \le \frac{\beta}{\rho(SJ)} ,$

where β is the minimal value of $\beta_{real}(x)$ or $\beta_{imag}(x)$ for x in [0,1], respectively. We shall call β the *SRK stability boundary* respectively for parabolic and hyperbolic problems.

It is our aim to construct methods with large SRK stability boundaries. In order to compare these boundaries of the various methods, we define *scaled* SRK boundaries. Suppose that the SRK method requires m evaluations of the righthand side function **f** and m* applications of the smoothing matrix S. Then we define the *scaled SRK stability boundaries*

$$\beta_f := \frac{\beta}{m}\ ,\quad \beta_S := \frac{\beta}{m^*}\ .$$

Finally, we remark that the application of conventional RK methods requires an estimate of the spectral radius of J, whereas the application of SRK methods requires information on the magnitude of the spectral radius of SJ. In practice, these estimates are usually obtained by a normal mode analysis. In the following examples this will be illustrated for a parabolic and hyperbolic model problem in the case where S is defined by {(2.6),(2.8)}.

Example 2.1. Consider the diffusion problem

$$\frac{\partial u}{\partial t} = d(u,x,t)\frac{\partial^2 u}{\partial x^2} + s(x,t),\ \ d(u,x,t) > 0,\ \ 0 \le t \le 1,\quad 0 \le x \le 1,$$

with Dirichlet boundary conditions. Standard symmetric finite difference discretization of the spatial derivatives on a uniform grid with mesh size Δ leads to a system of ODEs given by

$$\frac{dy_j}{dt} = \frac{d(y_j,j\Delta,t)}{\Delta^2}(y_{j+1} - 2y_j + y_{j-1}) + s(j\Delta,t),\ \ j=1, \dots , M\text{-}1;\ \ M := \frac{1}{\Delta},$$

where y_0=a(t) and y_M=b(t) are given boundary functions. In the implementation of SRK methods it is convenient to define differential equations at the boundary points. Therefore, we add the equations

$$\frac{dy_0}{dt} = a'(t),\ \frac{dy_M}{dt} = b'(t).$$

Suppose that the diffusion coefficient d(u,x,t) equals 1, then the Jacobian matrix J of the system of differential equations equals $4D/\Delta^2$ and applying a local mode analysis, it can be verified that

$$\lambda(J) = \frac{4\,\lambda(D)}{\Delta^2},\ \ \lambda(S) = \frac{T_k(1+2\lambda(D)) - 1}{2\lambda(D)k^2},\ \ \lambda(D) = -\frac{1}{2} + \frac{1}{2}\cos(\xi)\ ,\ \ \xi = \frac{j\pi\Delta}{1\,+\,2\Delta},\ \ j = 0, 1, \dots , M$$

so that

$$\text{(2.13)}\qquad \lambda(SJ) = \frac{T_k(1+2\lambda(D)) - 1}{2\lambda(D)k^2}\lambda(J) = 2\frac{T_k(1+2\lambda(D)) - 1}{k^2\Delta^2} = 2\ \frac{\cos(\xi) - 1}{k^2\Delta^2}\ .$$

From this expression we find that

$$\rho(SJ) \approx \frac{4}{k^2\Delta^2}\,.$$

In the case where d(u,x,t)≠1, we may use

$$\rho(SJ) \approx \frac{4\|\mathbf{d}(t)\|_\infty}{k^2\Delta^2}\ ,$$

where **d**(t) has components $d(y_j,j\Delta,t)$. Thus, when compared with $\rho(J)\approx 4\|\mathbf{d}(t)\|_\infty/\Delta^2$, we have a reduction of the spectral radius by a factor k^2.

Example 2.2. Consider the hyperbolic equation

$$\frac{\partial u}{\partial t} = a(x,t,u)\frac{\partial u}{\partial x}, \quad a(x,t,u) < 0, \quad 0 \le t \le 1, \quad 0 \le x \le 1,$$

with given boundary values b(t) at x=0. Semidiscretization by symmetric differences on a uniform grid $\{x_j = j\Delta\}$ leads to the system

$$\frac{dy_j}{dt} = \frac{a_j}{2\Delta}(y_{j+1} - y_{j-1}), \ j=1, \dots , M-1; \ a_j := a(j\Delta,t,y_j), \ M := \frac{1}{\Delta},$$

where y_0 is given. In order to compute y_0 and y_M, we add the equations

$$\frac{dy_0}{dt} = b'(t), \quad \frac{dy_M}{dt} = \frac{a_M}{2\Delta}(3y_M - 4y_{M-1} + y_{M-2}).$$

Suppose that the coefficient a(x,t,u) equals -1, then, ignoring the 'irregular' equation for y_M, we find by standard normal mode analysis

$$\lambda(J) = i\frac{\sin(\xi)}{\Delta}, \quad \lambda(D) = -\frac{1}{2} + \frac{1}{2}\cos(\xi),$$

so that

$$(2.14) \qquad \lambda(SJ) = \frac{T_k(1+2\lambda(D)) - 1}{2\lambda(D)k^2}\lambda(J) = \pm i\frac{T_k(1+2\lambda(D)) - 1}{k^2\Delta}\sqrt{\frac{1+\lambda(D)}{-\lambda(D)}}.$$

For large values of k, the extrema in the interval $-1 \le \lambda(D) \le 0$ of this function are assumed at the point which is approximately given by $(\cos(\pi/k)-1)/2$. This leads to

$$\rho(SJ) \approx \frac{2}{k^2\Delta}\frac{1}{\tan(\pi/2k)} \approx \frac{4}{k\pi\Delta}.$$

In the case where $a(x,t,u) \ne -1$, we shall use

$$\rho(SJ) \approx \frac{4\|\mathbf{a}(t)\|_\infty}{k\pi\Delta},$$

where $\mathbf{a}(t)$ has the components $a(j\Delta,t,y_j)$. Thus, in the hyperbolic case, we obtain, when compared with $\rho(J) \approx \|\mathbf{a}(t)\|_\infty/\Delta$, a reduction of the spectral radius by a factor $k\pi/4$.

2.3. The smoothing error

The smoothing error will be investigated by considering the error

$$(2.15) \qquad \sigma_{n+1} := h^{-1}[y_{n+1} - v_{n+1}],$$

where v_{n+1} denotes the numerical solution obtained by (2.3) with S=I and $y_n = v_n$, that is, the numerical solution obtained when in the nth step no smoothing is applied. In deriving an expression for σ_{n+1} we shall assume that for sufficiently smooth grid functions $\mathbf{u} = \mathbf{u}(\Delta)$ defined on a grid with mesh size Δ, the smoothed grid function $S\mathbf{u}(\Delta)$ converges to $\mathbf{u}$ as Δ tends to 0. Our goal is an expansion of the smoothing error in terms of powers of h and S-I.

From (2.3) it follows that

$$\sigma_{n+1} = \sum_{i=1}^{m} \left[B_i(S)\,\mathbf{f}(t_n + c_i h, \mathbf{Y}_i) - B_i(I)\,\mathbf{f}(t_n + c_i h, \mathbf{V}_i)\right],$$

$$\mathbf{Y}_i = y_n + h\sum_{j=1}^{i-1} A_{ij}(S)\,\mathbf{f}(t_n + c_j h, \mathbf{Y}_j), \quad \mathbf{V}_i = y_n + h\sum_{j=1}^{i-1} A_{ij}(I)\,\mathbf{f}(t_n + c_j h, \mathbf{V}_j), \quad i = 1, \dots , m.$$

For sufficiently differentiable righthand side functions we may write

$$Y_i - V_i = h \sum_{j=1}^{i-1} \Big[A_{ij}(S)f(t_n+c_jh,Y_j)-A_{ij}(I)f(t_n+c_jh,V_j)\Big] = h \sum_{j=1}^{i-1} \Big[[A_{ij}(S)-A_{ij}(I)]f(t_n+c_jh,V_j) + O(Y_j-V_j)\Big].$$

Introducing the Jacobian matrix $J_n := \frac{\partial}{\partial y} f(t_n,y_n)$, we may write

$$\begin{aligned}
\sigma_{n+1} &= \sum_{i=1}^{m} \Big[B_i(S)\, f(t_n + c_ih,V_i + Y_i - V_i) - B_i(I)\, f(t_n + c_ih,V_i)\Big] \\
&= \sum_{i=1}^{m} \Big[[B_i(S) - B_i(I)]\, f(t_n + c_ih,V_i) + B_i(S)J_n(Y_i - V_i) + O\big(h(Y_i - V_i) + (Y_i - V_i)^2\big)\Big] \\
&= \sum_{i=1}^{m} \Big[[B_i(S) - B_i(I)]\, \big(f(t_n,y_n) + hc_if_t(t_n,y_n) + hJ_n \sum_{j=1}^{i-1} A_{ij}(I)\, f(t_n,y_n)\big) \\
&\qquad + h\, B_i(S)J_n \sum_{j=1}^{i-1} [A_{ij}(S) - A_{ij}(I)]\, f(t_n,y_n)\Big] + O(h^2).
\end{aligned}$$

By using (2.3c) and the relations

(2.16) $$y(t_n) = y_n,\ \ y'(t_n) = f(t_n,y_n),\ \ y''(t_n) = f_t(t_n,y_n) + J_nf(t_n,y_n),$$

we obtain

(2.17) $$\sigma_{n+1} = \sum_{i=1}^{m} \Big[[B_i(S) - B_i(I)]\, \big(y'(t_n) + hc_iy''(t_n)\big) + h\, B_i(S)J_n \sum_{j=1}^{i-1} [A_{ij}(S) - A_{ij}(I)]\, y'(t_n)\Big] + O(h^2).$$

Thus, by defining the functions

(2.18) $$\alpha(x) := \sum_{i=1}^{m} [B_i(x) - B_i(1)],\quad \beta(x) := \sum_{i=1}^{m} c_i[B_i(x) - B_i(1)],\quad \gamma(x,y) := \sum_{i=1}^{m} B_i(x)y \sum_{j=1}^{i-1} [A_{ij}(x) - A_{ij}(1)],$$

the smoothing error can be represented in the compact form

(2.19) $$\sigma_{n+1} = \alpha(S)\, y'(t_n) + h\beta(S)\, y''(t_n) + h\gamma(S,J_n)\, y'(t_n) + O(h^2).$$

We remark that in the case where the matrices J_n and S commute this expression can be written in the form

(2.20) $$\sigma_{n+1} = \alpha(S)\, y'(t_n) + h\beta(S)\, y''(t_n) + h\gamma(S,1)\, J_ny'(t_n) + O(h^2).$$

We shall analyse the magnitude of the smoothing error by applying a normal mode analysis, that is, we substitute harmonic data for $y'(t_n)$ and $y''(t_n)$, and since the matrices J_n and S commute on the space spanned by harmonic data, we may use (2.20) as our starting point.

In this paper, we shall concentrate on SRK methods with smoothing matrices defined by {(2.6),(2.8)} and on problems where the low frequencies in $y'(t_n)$ and $y''(t_n)$ dominate. As was already observed in Section 2.3, the eigenvalues of such smoothing matrices lie in the interval [0,1]. Hence, the magnitude of the polynomials $\alpha(x)$, $\beta(x)$ and $\gamma(x,1)$ on the interval [0,1] determines the magnitude of the smoothing error σ_{n+1}. Since, by substituting harmonic data, it can be shown that the low frequencies correspond to eigenvalues of S close to 1, it follows that the low frequencies can be removed from the smoothing error by choosing the polynomials $\alpha(x)$, $\beta(x)$ and $\gamma(x,1)$ such that they possess a high-order zero at x=1 (notice that, by virtue of definition (2.18), $\alpha(x)$, $\beta(x)$ and $\gamma(x,1)$ already have a zero of order 1 at x=1). However, for larger values of k there will also be low frequencies corresponding to zero eigenvalues of S (see the discussion in Section 2.3). Thus, we should be prepared that the smoothing error will be substantial for large values of k and that, in view of (1.2), it cannot be reduced by specific choices of the polynomials B_i and A_{ij}.

Finally, we remark that in the two-dimensional case, where we set $S=S_1S_2$ with S_1 and S_2 respectively generated by difference matrices D_1 and D_2, similar results are obtained.

3. ONE-STAGE SRK METHODS

In this section we consider the case m=1, so that (2.3) reduces to

$$(3.1) \qquad y_{n+1} = y_n + h\, B_1(S)\, f(t_n,y_n).$$

Notice that these methods may also be considered as RHS methods generated by the Forward Euler method and employing the smoothing matrix $B_1(S)$.

In view of (2.9) we write $B_1(x)=xB(x)$, so that

$$\alpha(x) = xB(x) - B(1), \ \ \beta(x) = \gamma(x,y) = 0, \ \ R(z,x) = 1 + zB(x).$$

Since $\Lambda(S)$ is contained in [0,1], the variable x in these polynomials is restricted to the interval [0,1].

Evidently, the range of the complex variable z in R(z,x) is not allowed to contain purely imaginary values, otherwise R cannot be bounded by 1. This excludes the case of hyperbolic equations which would lead to imaginary values in the spectrum of J and therefore in the spectrum of hSJ. Therefore, our considerations will be restricted to the construction of methods for *parabolic* equations.

3.1. Parabolic SRK methods

Requiring that the unsmoothed method is consistent, we find that $B_1(1)=1$. It is convenient to present $B_1(x)$ in the form

$$B_1(x) = 1 - (1 - x)^p[1 + xC(x)], \ \ p > 1,$$

where C(x) is a polynomial such that $B_1(x)\geq 0$ in [0,1]. Evidently, $B_1(x)$ vanishes at x=0 and equals 1 at x=1 for all polynomials C(x). From (2.20) we derive that the smoothing error of (3.1) is given by

$$(3.2) \qquad \sigma_{n+1} = \ - (I - S)^p[I + SC(S)]y'(t_n) + O(h^2), \ \ p > 1.$$

In the real (z,x)-plane the stability polynomial of (3.1) has the stability region

$$\mathbb{S} = \{(z,x)\colon -2 \leq \frac{zB_1(x)}{x} \leq 0, \ \ 0 \leq x \leq 1\}.$$

For parabolic problems where $\Lambda(SJ)$ is negative, the stability of the method is determined by

$$h \leq \frac{\beta}{\rho(SJ)}, \quad \beta := \min\left\{\frac{2x}{B_1(x)}\right\}, \quad B_1(x) \geq 0, \quad 0 \leq x \leq 1.$$

Notice that the value of β can never exceed 2 (e.g., by setting x=1 we find that $\beta\leq 2$).

Theorem 3.1. *Let*

$$(3.3) \qquad B_1(x) = 1 - (1 - x)^p[1 + qx], \ \ p > 1,$$

where q *is a constant, then the SRK boundary* β *defined by* (3.3) *is maximal for* q=p.

Proof. From (3.3) it easily follows that in the interval [0,1] the values of $2x/B_1(x)$ are increasing functions of q. Hence, the stability boundary is larger as q is larger. However, $B_1(x)$ should be nonnegative in [0,1], and since $B_1(0)=0$, we should require that q is such that $B_1'(0)\geq 0$. From

$$B_1'(x) = [p - q + q(p + 1)x](1 - x)^{p-1}$$

it follows that the largest possible SRK boundary β is obtained for q=p, provided that the condition $B_1(x)\geq 0$ is satisfied on $0\leq x\leq 1$. Since $B_1'(x)\geq 0$ in [0,1] for q=p, the assertion of the theorem is proved. □

Below, we present the methods generated by Forward Euler employing smoothing matrices S, (2I-S)S, and $B_1(S)$ where B_1 is defined by (3.3) with p=q=2 and p=q=3. For future reference, we also specify the corresponding stability conditions (the smoothing errors can be derived from (3.2) by replacing C(S) with p).

(3.4) $$y_{n+1} = y_n + hSf(t_n,y_n),\ \ h \le \frac{\beta}{\rho(SJ)},\ \ \beta = \beta_f = \beta_S = 2.$$

(3.5) $$y_{n+1} = y_n + h(2I - S)Sf(t_n,y_n),\ \ h \le \frac{\beta}{\rho(SJ)},\ \ \beta = \beta_f = 1,\ \ \beta_S = \frac{1}{2}.$$

(3.6) $$y_{n+1} = y_n + h\left(3I - 2S\right) S^2 f(t_n,y_n),\ \ h \le \frac{\beta}{\rho(SJ)},\ \ \beta = \beta_f = \frac{16}{9},\ \ \beta_S = \frac{16}{27}.$$

(3.7) $$y_{n+1} = y_n + h\left(6I - 8S + 3S^2\right) S^2 f(t_n,y_n),\ \ h \le \frac{\beta}{\rho(SJ)},\ \ \beta = \beta_f = \frac{68 - 5\sqrt{10}}{36} \approx 1.45,\ \ \beta_S = \frac{\beta}{4}.$$

4. TWO-STAGE METHODS

In this section, we consider the two-stage method

(4.1a) $$y_{n+1} = y_n + h\, B_1(S)\, f(t_n,y_n) + h\, B_2(S)\, f(t_n + c_2h, y_n + hA_{21}(S)\, f(t_n,y_n)),$$

with

(4.1b) $$B_1(x) = x(b_1 + b_2x),\ \ B_2(x) = x(b_3 + b_4x),\ \ A_{21}(x) = x(a_1 + a_2x).$$

From (2.20) we obtain the smoothing error

(4.2) $$\sigma_{n+1} = [S - I][b_1 + b_2 + b_3 + b_4 + (b_2 + b_4)S]\, y'(t_n)$$
$$+ c_2h[S - I][b_3 + b_4 + b_4S]\, y''(t_n) + h[S - I]S[b_3 + b_4S][a_1 + a_2 + a_2S]\, J_ny'(t_n) + O(h^2).$$

We analyse in the following subsections SRK methods for both parabolic and hyperbolic equations.

4.1. Parabolic SRK methods

As a reference method, we give the RHS method generated by a two-stage RK method with maximal real stability boundary:

(4.3) $$y_{n+1} = y_n + hSf(t_n + \tfrac{1}{8}h, y_n + \tfrac{1}{8}hSf(t_n,y_n)),\ \ h \le \frac{\beta}{\rho(SJ)},\ \ \beta = 8,\ \ \beta_f = \beta_S = 4.$$

Its smoothing error is of the form

(4.2') $$\sigma_{n+1} = [S - I]\left\{ y'(t_n) + \tfrac{1}{8}hy''(t_n) + \tfrac{1}{8}hSJ_ny'(t_n) \right\} + O(h^2).$$

In this section we shall try to reduce the smoothing error of two-stage methods and, at the same time, to maximize the real stability boundary. By setting

(4.4a) $$b_1 + b_2 + b_3 + b_4 = 1,$$

in (4.1), we achieve that we have first-order time accuracy, and by setting

(4.4b) $$b_2 + b_4 = -1,$$

the smoothing error assumes the form

(4.2") $$\sigma_{n+1} = -[S - I]\left\{ (S - I)y'(t_n) - c_2h[b_3 + b_4 + b_4S]\, y''(t_n) - hS[b_3 + b_4S][a_1 + a_2 + a_2S]\, J_ny'(t_n) \right\} + O(h^2),$$

where a_1, a_2, b_3 and b_4 are free parameters. The stability polynomial corresponding to the condition (4.4) assumes the form

$$(4.5)\qquad R(z,x) = 1 + [2 - x]\, z + (a_1 + a_2x)(b_3 + b_4x)z^2.$$

First of all we observe that for x=0 this polynomial reduces to $R(z,x)=1+2z+a_1b_3z^2$. It is easily shown that this polynomial has optimal real stability boundary 4 which is achieved for $a_1b_3=1/2$. Hence, the parabolic SRK stability boundary cannot exceed the value 4.

Let us write (4.5) in the form

$$(4.5')\qquad R(z,x) = 1 + [2 - x]\, z + c(x)\big[[2 - x]\, z\big]^2, \quad c(x) := \frac{(a_1 + a_2x)(b_3 + b_4x)}{(2 - x)^2}\ .$$

It is easily verified that for a fixed value of x the real stability boundary $\beta_{real}(x)$ of R is given by

$$\beta_{real}(x) := \frac{1}{(2 - x)c(x)}\,,$$

provided that $c(x)\geq 1/8$.

One possibility to get the maximal value 4 is choosing the free parameters a_1, a_2, b_3 and b_4 such that the coefficient function c(x) equals 1/8. Then, $\beta_{real}(x)=8/(2-x)$, so that the parabolic SRK stability boundary equals the maximal value 4 indeed. It is easily verified that the condition c(x)=1/8 is satisfied if

$$a_1 = \frac{1}{2b_3}, \quad a_2 = \frac{-1}{4b_3}, \quad b_1 = 2 - b_3, \quad b_2 = -\frac{1}{2}(2 - b_3), \quad b_4 = -\frac{1}{2}b_3.$$

The corresponding method reads

$$y_{n+1} = y_n + \frac{1}{2}h(2I - S)S\Big[[2 - b_3]f(t_n\,,y_n) + b_3f(t_n + \frac{1}{4b_3}h, y_n + \frac{h}{4b_3}(2I - S)Sf(t_n\,,y_n))\Big].$$

possessing the smoothing error

$$(4.6)\qquad \sigma_{n+1} = -\,[S - I]^2\Big\{\, y'(t_n) + \frac{1}{8}hy''(t_n) + \frac{1}{8}(2I - S)ShJ_ny'(t_n)\Big\} + O(h^2)$$

for all nonzero values of the free parameter b_3. The free parameter b_3 cannot be used for a further reduction of the smoothing error. Therefore, we chose in our experiments $b_3=2$ to obtain

$$(4.7)\qquad y_{n+1} = y_n + h(2I - S)Sf(t_n + \frac{1}{8}h, y_n + \frac{1}{8}h(2I - S)Sf(t_n\,,y_n)), \ h \leq \frac{\beta}{\rho(SJ)}, \ \beta = 4, \ \beta_f = 2, \ \beta_S = 1.$$

Another possibility is the choice $(2-x)c(x)=1/4$, so that $\beta_{real}(x)$ equals 4 for all x. This leads to

$$a_1 = \frac{1}{2b_3}, \quad a_2 = 0, \quad b_1 = 2 - b_3, \quad b_2 = -\frac{1}{2}(2 - b_3), \quad b_4 = -\frac{1}{2}b_3,$$

to obtain the method

$$y_{n+1} = y_n + \frac{1}{2}h(2I - S)S\Big[[2 - b_3]f(t_n\,,y_n) + b_3f(t_n + \frac{h}{2b_3}, y_n + \frac{h}{2b_3}Sf(t_n\,,y_n))\Big],$$

with smoothing error

$$(4.8)\qquad \sigma_{n+1} = -\,[S - I]\Big\{\,(S - I)y'(t_n) + \frac{1}{4}h[S - I]\, y''(t_n) - \frac{1}{4}hS[2I - S]J_ny'(t_n)\Big\} + O(h^2)$$

for all nonzero values of the free parameter b_3. Again setting $b_3=2$ the SRK method reduces to

$$(4.9)\qquad y_{n+1} = y_n + h(2I - S)Sf(t_n + \frac{1}{4}h, y_n + \frac{1}{4}hSf(t_n\,,y_n)), \ h \leq \frac{\beta}{\rho(SJ)}, \ \beta = 4, \ \beta_f = 2, \ \beta_S = \frac{4}{3}.$$

4.2. Hyperbolic SRK methods

The RHS method generated by a two-stage RK method with maximal imaginary stability boundary is given by

$$(4.10) \qquad y_{n+1} = y_n + hSf(t_n + h, y_n + hSf(t_n, y_n)), \; h \leq \frac{\beta}{\rho(SJ)}, \; \beta = 1, \; \beta_f = \beta_S = \frac{1}{2},$$

with smoothing error

$$(4.11) \qquad \sigma_{n+1} = [S - I]\left\{ y'(t_n) + hy''(t_n) + hSJ_n y'(t_n) \right\} + O(h^2).$$

As in the parabolic case, we impose condition (4.4) to ensure first-order accuracy and a smoothing error of the form (4.2"). From the corresponding stability function (4.5) we deduce that for x=0 the imaginary stability boundary is at most 1/2, so that the hyperbolic SRK stability boundary of two-stage methods is also limited by 1/2.

Proceeding as for parabolic equations, we first determine the imaginary stability boundary $\beta_{imag}(x)$ for fixed x. This yields

$$\beta_{imag}(x) := \frac{\sqrt{2c(x) - 1}}{(2 - x)c(x)}.$$

Requiring that $\beta_{imag}(x) \geq 1/2$ on $0 \leq x \leq 1$, we find that

$$(4.12) \qquad x(4 - x)c^2(x) - 4(c(x) - 1)^2 \geq 0, \; 0 \leq x \leq 1.$$

This inequality immediately shows that we should choose c(0)=1, i.e., $a_1b_3=4$. In order to simplify the derivations we set $a_2=0$, with the advantage that only three applications of S are involved. A simple calculation reveals that (4.12) can be written as

$$(a_1b_4)^2 \leq 4\,\frac{4 - x}{x},$$

so that we should choose a_1b_4 such that $(a_1b_4)^2$ is less than or equal to 12. We shall use the freedom in choosing a_1b_4 for minimizing the second-order time error constant. By setting S=I in (4.1), this error constant is easily seen to be

$$C_2 := (a_1 + a_2)(b_3 + b_4) - \frac{1}{2} = \frac{7}{2} + a_1b_4.$$

This is justified by observing that in the case of hyperbolic problems the stability condition prescribes a timestep h of $O(\Delta)$, so that the total global discretization error is of $O(h+\Delta^2+\sigma)$ showing that the time discretization error is dominating. From the expression for C_2 it follows that the error constant C_2 assumes its minimal value 0.036 for $a_1b_4=-\sqrt{12}$. The corresponding SRK method assumes the form

$$(4.13) \qquad \begin{aligned} &y_{n+1} = y_n + \frac{1}{a_1}hS\Big[\Big(2a_1 - 4 + (\sqrt{12} - a_1)\,S\Big)\; f(t_n,y_n) + (4 - \sqrt{12}\,S)\; f(t_n + a_1h, y_n + a_1hS\; f(t_n,y_n))\Big], \\ &h \leq \frac{\beta}{\rho(SJ)}, \; \beta = \frac{1}{2}, \; \beta_f = \frac{1}{4}, \; \beta_S = \frac{1}{6}. \end{aligned}$$

In our numerical experiments we always used the value $a_1=1/2$.

The method (4.13) possesses for all nonzero values of a_1 a minimized local time error constant and smoothing error

$$(4.14) \qquad \sigma_{n+1} = - [S - I]\left\{ (S - I)y'(t_n) - h[4 - \sqrt{12} - \sqrt{12}\,S]\; y''(t_n) - hS[4 - \sqrt{12}\,S]\; J_n y'(t_n) \right\} + O(h^2).$$

5. NUMERICAL EXPERIMENTS

In this section we test the SRK methods constructed in the preceding sections. As in the Examples 2.1 and 2.2, we use standard, symmetric semi-discretizations that are second-order accurate with respect to the spatial meshsize Δ for generating the systems of ODEs. Hence, if the global time integration error equals p, then we can achieve a fully discretized scheme of which the total global error, that is, the sum of the spatial discretization error, the time discretization error and the smoothing error is given by $O(h^p+\Delta^2+\sigma)$, where σ denotes an upper bound for all local smoothing errors σ_{n+1}. For convenience, the smoothing error σ_{n+1} and the stability boundaries for the various SRK methods are summarized in Table 5.1.

Table 5.1. Smoothing errors and stability boundaries of SRK methods.

Problem	Method	σ_{n+1}	β	β_f	β_s
Parabolic	(3.4)	$-(I-S)y'(t_n) + O(h^2)$	2	2	2
	(3.5)	$-(I-S)^2y'(t_n) + O(h^2)$	1	1	0.5
	(3.6)	$-(I-S)^2[I + 2S]y'(t_n) + O(h^2)$	1.78	1.78	0.59
	(3.7)	$-(I-S)^3[I + 3S]y'(t_n) + O(h^2)$	1.45	1.45	0.36
	(4.3)	$[S-I]\,[y'(t_n) + O(h)] + O(h^2)$	8	4	4
	(4.7)	$-[S-I]^2[y'(t_n) + O(h)] + O(h^2)$	4	2	1
	(4.9)	$-[S-I]\;[(S-I)y'(t_n) + O(h)] + O(h^2)$	4	2	1.33
Hyperbolic	(4.10)	$[S-I]\,[y'(t_n) + O(h)] + O(h^2)$	1	0.50	0.50
	(4.13)	$-[S-I]\,[(S-I)y'(t_n) + O(h)] + O(h^2)$	0.50	0.25	0.16

In our experiments, we choose the stepsize h of the SRK method as large as the stability condition (2.12) allows. Setting $k=2^q$ and using the relation $\rho(J)\approx k^2\rho(SJ)$ for parabolic problems (see Example 2.1) and the relation $\rho(J)\approx k\pi\rho(SJ)/4$ for hyperbolic problems (see Example 2.2), we obtain, respectively, the maximal stepsizes

$$(5.1)\qquad h_{par} = \frac{\beta}{\rho(SJ)} = \frac{\beta 4^q}{\rho(J)} = \frac{\beta 4^q \Delta^2}{R}, \quad h_{hyp} = \frac{\beta}{\rho(SJ)} = \frac{\beta\pi 2^q}{4\rho(J)} = \frac{\beta\pi 2^q\Delta}{4R},$$

where β is the SRK stability boundary and R is a constant determined by the problem to be solved. Hence, we find that the total global error for parabolic and hyperbolic problems is respectively given by

$$(5.2)\qquad \varepsilon_{par} = \sigma + O(h) + O(\Delta^2) = \sigma + O(\Delta^2),\quad \varepsilon_{hyp} = \sigma + O(h) + O(\Delta) = \sigma + O(\Delta).$$

The accuracy is measured by the number A of correct decimal digits at the end point of the integration interval, i.e., by

$$(5.3)\qquad A := -\log(\|\text{error at the end point }\|_\infty),$$

and the computational costs are estimated by the total number of scalar righthand side evaluations. From the expressions (5.1) for the timestep h it follows that, for a given (one-dimensional) problem, this number is given by

$$(5.4)\qquad N_{par} := \frac{mC}{\beta 4^q \Delta^3},\quad N_{hyp} := \frac{mC}{\beta\pi 2^q \Delta^2},$$

where C is a constant only depending on the problem at hand.

In order to achieve that the results produced by the various methods can be compared, we shall choose for each method the spatial mesh such that N is roughly constant. To be more precise, a method with SRK stability boundary β is applied with grid parameters

$$(5.5)\qquad \Delta_{par} \approx \delta\sqrt[3]{\frac{2}{\beta}},\quad h_{par} \approx \frac{\beta 4^q(\Delta_{par})^2}{R}$$

in the case of parabolic problems, and with

$$(5.6)\qquad \Delta_{hyp} \approx \delta\sqrt{\frac{1}{\beta}}\,,\quad h_{hyp} \approx \frac{\beta\pi 2^q \Delta_{hyp}}{4R}$$

in the hyperbolic case. In (5.5), δ is the meshsize for parabolic SRK methods with stability boundary β=2 (e.g., the RHS reference method (3.4)), and in (5.6) δ is the meshsize for hyperbolic SRK methods with stability boundary β=1 (e.g., the RHS method (4.10)). Thus, δ and q are the new parameters to control the accuracy and the computational costs.

5.1. A nonlinear diffusion problem

Consider the diffusion problem

$$(5.7)\qquad \frac{\partial u}{\partial t} = e^u \frac{\partial^2 u}{\partial x^2} + u(9e^u - 1),\quad 0 \le t \le 1,\quad 0 \le x \le 1,$$

with exact solution $u(x,t)=e^{-t}\sin(3x)$. For this problem we have R=4e.

The errors obtained for a few values of the spatial gridsize δ are listed in Table 5.2. The subsequent rows in Table 5.2 correspond to the methods specified in the first column. Each of the other columns contain results which are roughly equally expensive (about the same N_{par} value), facilitating an easy comparison (the most accurate results in each column are indicated in bold).

Table 5.2. Numbers of correct decimal digits obtained for problem (5.7).

(δ^{-1}, q)	(29,3)	(29,4)	(58,4)	(58,5)	(116,5)	(116,6)
(3.4)	1.93	1.35	2.01	1.37	2.03	1.38
(3.5)	2.43	1.65	**3.27**	1.72	**3.28**	1.74
(3.6)	2.60	1.57	3.23	1.61	3.19	1.62
(3.7)	2.90	2.00	2.81	2.10	2.67	2.13
(4.3)	2.43	1.90	2.61	1.95	2.66	1.96
(4.7)	2.89	2.29	2.50	2.15	2.43	2.12
(4.9)	**3.21**	**2.61**	2.74	**2.46**	2.63	**2.40**
$N_{par} \approx$	2120	530	4240	1060	8480	2120

This table clearly demonstrates that the smoothing error can be considerably reduced without increasing the computational costs by using more sophisticated methods than the straightforward RHS method (3.4). In this example the methods (3.5) and (4.9) produce the most accurate results.

5.2. A nonlinear convection problem

Consider the convection problem (cf. Example 2.2)

$$(5.8)\qquad \frac{\partial u}{\partial t} = -u^2\frac{\partial u}{\partial x} + g(t,x),\quad 0 \le t \le 1,\quad 0 \le x \le 1,$$

where g is such that the exact solution is given by $u(x,t)=5+\sin(t)\cos(3\pi x)$. For this problem we have R=36.

Table 5.3 presents the analogue of Table 5.2. It shows that the accuracy of the RHS method (4.10) drops much more when q is increased than the accuracy of the SRK method (4.13).

Table 5.3. Numbers of correct decimal digits obtained for problem (5.8).

(δ^{-1}, q)	(31,2)	(31,3)	(62,3)	(62,4)	(124,4)	(124,5)
(4.10)	1.36	0.64	1.72	0.69	1.90	0.70
(4.13)	1.09	0.98	1.61	1.28	1.83	1.37
$N_{hyp} \approx$	22100	11050	44200	22100	88400	44200

REFERENCES

[1] Butcher, J.C. (1987): The numerical analysis of ordinary differential equations, Runge-Kutta and general linear methods, Wiley, New York.

[2] Houwen, P.J. van der (1977): Construction of integration formulas for initial-value problems, North-Holland, Amsterdam.

[3] Houwen, P.J. van der, Boon, C. & Wubs, F.W. (1988): Analysis of smoothing matrices for the preconditioning of elliptic difference equations, Z. Angew. Math. Mech. **68**, 3-10.

[4] Houwen, P.J. van der, Sommeijer, B.P. & Wubs, F.W. (1990): Analysis of smoothing operators in the solution of partial differential equations by explicit difference schemes, Appl. Numer. Math. **6**, 501-521.

[5] Jameson, A. (1983): The evolution of computational methods in aerodynamics, J. Appl. Mech. **50**, 1052-1076.

[6] Lerat, A. (1979): Une class de schémas aux différence implicites pour les systemes hyperboliques de lois de conservation, C.R. Acad. Sc. Paris, t. **288** (Serie A), 1033-1036.

[7] Strehmel, K. (1981): Stability properties of adaptive Runge-Kutta methods (in German), Z. Angew. Math. Mech. **61**, 253-260.

[8] Turkel, E. (1985): Acceleration to a steady state for the Euler equations, in: Numerical methods for the Euler equations of fluid dynamics, SIAM Publications, Philadelphia, 218-311.

[9] Wubs, F.W. (1986): Stabilization of explicit methods for hyperbolic partial differential equations, Int. J. Numer. Methods Fluids **6**, 641-657.