INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 30, 271–290 (1990)

ITERATED θ-METHOD FOR HYPERBOLIC EQUATIONS

P. J. VAN DER HOUWEN AND B. P. SOMMEIJER

CWI, Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

SUMMARY

The iterated θ-methods employing residue smoothing for finding both steady state and time-accurate solutions of semidiscrete hyperbolic differential equations are analysed. By the technique of residue smoothing the stability condition is considerably relaxed, so that larger time steps are allowed which improves the efficiency of the method. The additional computational effort involved by the explicit smoothing technique used here is rather low when compared with its stabilizing effect. However, in the case where time-accurate solutions are desired, the overall accuracy may be decreased. This paper investigates the effect of residue smoothing on both the stability and accuracy, and presents a number of explicitly given methods based on the iterated implicit midpoint rule ($\theta = 1/2$). Numerical examples confirm the theoretical results.

INTRODUCTION

In References 2 and 3 function iteration methods for solving the implicit relations associated with implicit linear multistep methods were studied in the case of semidiscrete parabolic differential equations. It was shown that iterated multistep methods could be stabilized considerably by using residue smoothing techniques based on smoothing matrices. The impressive parabolic stability boundaries derived in these papers tempted us to study residue smoothing techniques in iterated multistep methods for semidiscrete hyperbolic equations. However, a first investigation revealed that the hyperbolic case is much more complicated than the parabolic case and therefore we decided to start with the relatively simple case of an iterated one-step method; in fact, we chose the iterated θ-method. In this paper, we show that, in hyperbolic schemes, residue smoothing may also relax the stability conditions substantially. It turned out that the smoothers used in parabolic problems are not suitable in the hyperbolic case, so that we have to construct new smoothing matrices. It should be remarked that residue smoothing in solving hyperbolic problems has already been used by Lerat,[8] Jameson[4] and Turkel,[10] but these authors employ quite different, usually implicit, smoothing operators.

The numerical schemes obtained in this paper are completely explicit and, by virtue of their simple structure, they vectorize extremely well on vector computers. They can be used for finding both steady state and time-accurate solutions of semidiscrete hyperbolic differential equations. The methods are illustrated by integrating the hyperbolic initial-boundary-value problem

$$\frac{\partial u}{\partial t} = a(x, t, u)\frac{\partial u}{\partial x}, \quad u(x, t_0) = g(x), \quad u(0, t) = b(t), \quad a(x, t, u) < 0, \quad 0 \leqslant x \leqslant 1, \quad t \geqslant t_0 \quad (1)$$

Suppose that this problem is semidiscretized by symmetric differences on a uniform grid

0029–5981/90/100271–20$10.00

Received 26 July 1988
Revised 11 January 1990

$\{x_j = j\Delta x\}$ and let the resulting system of differential equations be given by

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

where $y_0 = b(t)$ and $y_j = y_j(t)$ approximates $u(x_j, t)$. The last equation in this system asks for y_M. In order to compute this component we add the equation

$$\frac{dy_M}{dt} = \frac{a_M}{2\Delta x}(3y_M - 4y_{M-1} + y_{M-2}) \tag{2b}$$

It should be remarked that the above symmetric semidiscretization is not necessarily a suitable discretization of the space derivative for all initial functions $g(x)$ and coefficient functions $a(x, t, u)$. However, if these functions are such that the exact solution $u(x, t)$ is smooth, then symmetric spatial discretizations are justified. An important class of 'real-life' problems described by hyperbolic initial-boundary-value problems which do have smooth solutions are the shallow water problems (see, e.g. Reference 9 and the references cited there). In fact, this work was carried out at CWI as part of the VECPARCOMP project for designing numerical three-dimensional shallow water models on vector and parallel computers. Moreover, for such problems the method of lines approach (i.e. the separate treatment of spatial discretization and time integration) is quite usual and will be followed in this paper.

In the Appendix to this paper, the final algorithm for the time integration of (2) together with various smoothing matrices are explicitly given.

THE ITERATED θ-METHOD

Let the semidiscrete hyperbolic differential equation be given by the system of ordinary differential equations

$$\frac{dy(t)}{dt} = f(t, y(t)) \tag{3}$$

and consider the so-called θ-method with stepsize $h := t_{n+1} - t_n$ (see, e.g. Reference 1, p. 199):

$$y_{n+1} = y_n + hf(t_n + \theta(t_{n+1} - t_n), y_n + \theta(y_{n+1} - y_n)), \quad 0 < \theta \leq 1 \tag{4}$$

This method requires in each step point $t = t_{n+1}$ solving the equation

$$R_n(t, y) := y - y_n - hf(t_n + \theta(t - t_n), y_n + \theta(y - y_n)) = 0 \tag{5}$$

for y. One way to solve this equation is by means of the one-point function iteration method

$$y^{(0)} = y_n; \quad y^{(j)} = y^{(j-1)} - r^{(j)} R_n(t^{(j-1)}, y^{(j-1)})$$
$$t^{(0)} = t_n; \quad t^{(j)} = t^{(j-1)} - r^{(j)}[t^{(j-1)} - t_n - h]$$

where $j = 1, \ldots, m$ and $\{r^{(j)}\}$ are relaxation parameters. When we accept $y^{(m)}$ as an approximation to the exact solution η of (5), that is we set $y_{n+1} = y^{(m)}$, then this scheme may be interpreted as an m-stage, one-step Runge–Kutta method. By applying the well-developed Runge–Kutta theory, the relaxation parameters can be determined in such a way that y_{n+1} has order of accuracy 1 or 2, and possesses optimal stability for hyperbolic problems. Unfortunately, it turns out that, in order to achieve sufficient stability for a realistic integration step, the number of iterations should be relatively large and hence a lot of computational effort is required. In order to save computing time, we consider the following modification:

$$
\begin{aligned}
y^{(0)} = y_n; \quad & y^{(j)} = y^{(j-1)} - r^{(j)} S R_n(t^{(j-1)}, y^{(j-1)}) \\
t^{(0)} = t_n; \quad & t^{(j)} = t^{(j-1)} - r^{(j)}[t^{(j-1)} - t_n - h]
\end{aligned}
\tag{6}
$$

where $j = 1, \ldots, m$. Here, S is a smoothing matrix which is defined by a polynomial of degree k of a difference matrix D, i.e.

$$S = S_k(D) \tag{7a}$$

where D is some (possibly rough) approximation to the normalized Jacobian matrix of the right-hand-side function:

$$D \approx \frac{J}{\rho}, \quad J := \frac{\partial f}{\partial y}(t_n + \theta(t_{n+1} - t_n), y_n + \theta(\eta - y_n)) \tag{7b}$$

with ρ denoting the spectral radius of the matrix J. The *smoothing polynomial* $S_k(x)$ is required to satisfy the condition $S_k(0) = 1$, so that S approximates the identity matrix in the space spanned by eigenvectors of D with eigenvalues close to the origin. Examples of smoothing polynomials are

$$S_k(x) = \frac{T_{k+1}(1 + 2x) - 1}{2(k+1)^2 x}, \quad S_{2k}(x) = \frac{T_{k+1}(1 + 2x^2) - 1}{2(k+1)^2 x^2}, \quad S_{2k}(x) = \frac{U_{2k}(\sqrt{1 + x^2})}{2k + 1} \tag{8}$$

Here, T_m and U_m denote Chebyshev polynomials of the first and second kind. For special values of k these polynomials allow an extremely efficient implementation of the corresponding smoothing matrices (cf. Reference 2). The first family of these polynomials is appropriate in parabolic problems.

The scheme (6) may be interpreted as an m-stage, one-step Runge–Kutta method in which the Runge–Kutta parameters are replaced by matrices. Thus, the iterated θ-method can be represented by a Butcher array. For example, the Butcher arrays for $m = 1$ and $m = 2$ are given by

$$
\begin{array}{c|c}
0 & 0 \\
\hline
 & r^{(1)}S
\end{array}\,, \qquad
\begin{array}{c|cc}
0 & 0 & 0 \\
\theta r^{(1)} & \theta r^{(1)} S & 0 \\
\hline
 & (r^{(1)} - r^{(1)} r^{(2)} S) S & r^{(2)} S
\end{array}
$$

In practice, however, the representation (6) is more suited for implementation.

Amplification polynomial

In view of the special form (6), the most obvious approach is to choose the relaxation parameters such that the iteration error is rapidly decreased in magnitude. Since the iteration error in (6) is approximately given by

$$e_{n+1} := \eta - y_{n+1} \approx P_m(S[I - \theta Z]) e_n, \quad P_m(x) := \prod_{j=1}^{m} [1 - r^{(j)} x], \quad Z := hJ \tag{9}$$

we should choose the *amplification polynomial* $P_m(x)$ of the θ-method appropriately. Usually, the iteration error e_n is dominated by eigenvectors of $S[I - \theta Z]$ with eigenvalues close to 1. This suggests that we should choose $P_m(x)$ such that it is small in magnitude in the neighbourhood of $x = 1$. Ideally, we should minimize $P_m(x)$ on the set of eigenvalues of $S[I - \theta Z]$ which are close to 1. However, owing to the introduction of the polynomial S_k, these eigenvalues are located on a complicated curve in the complex plane and it seems unlikely that we can exploit its particular form. Therefore, we consider the spectrum of $S[I - \theta Z]$ as an arbitrary set of points in the complex plane, and, by applying Zarantonello's lemma (cf. Reference 11), we see that, as far as damping of the iteration error is concerned, we cannot do better than concentrating all zeros of $P_m(x)$ at a fixed point in the centre of the region where we want $P_m(x)$ to be small in magnitude. Obviously, this leads us to equal values for $r^{(j)}$ and, since at least one relaxation parameter should be 1, we find $r^{(j)} = 1$ for all j, so that

$$P_m(x) := [1 - x]^m \tag{10}$$

In the next section, where we investigate the linear stability of the iterated θ-method, we shall see that this polynomial is also obtained by solving the order conditions derived from the stability polynomial of the iterated θ-method.

Stability polynomial

In order to derive the stability polynomial for the iterated θ-method, we have to establish a (approximate) relation between y_{n+1} and y_n when the method is applied to the linear test equation $y' = Jy = h^{-1}Zy$. From (9) we have

$$\eta - y_{n+1} = P_m(S[I - \theta Z])[\eta - y_n] \tag{9'}$$

and from (5) we find that

$$\eta = [I - \theta Z]^{-1}[I + (1 - \theta)Z]y_n$$

Substitution of η into (9') and using (7a) yields

$$y_{n+1} = R(D, Z)y_n, \quad R(x, z) := \frac{1 + [1 - \theta - P_m(S_k(x)[1 - \theta z])]z}{1 - \theta z} \tag{11}$$

$R(D, Z)$ will be called the *stability matrix* and $R(x, z)$ the *stability polynomial*. Notice that $R(x, z)$ is a polynomial in both x and z. We remark that in the ideal case (the so-called *model situation*) where D equals the normalized Jacobian, i.e. if $D = \rho^{-1}J = (h\rho)^{-1}Z$, the stability polynomial $R(x, z) = R((h\rho)^{-1}z, z)$ is a polynomial of z alone. This 'simplified' stability polynomial plays a central role in our (linear) stability considerations.

Unfortunately, when using the smoothing polynomials (8), the method generated by the amplification polynomial (10) has poor stability properties. As an alternative and more successful approach, we choose the smoothing polynomial in such a way that the resulting stability polynomial of the θ-method is suitable for integrating hyperbolic equations. Since stability polynomials govern not only the stability but to some extent also the overall accuracy of one-step methods we shall derive accuracy and stability conditions at the same time. In this approach, we may profit from the many results available in the literature on stability polynomials for hyperbolic equations (e.g. References 5, 6 and 7). In this connection, we observe that if we succeed in identifying $R((h\rho)^{-1}z, z)$ with a given stability polynomial with constant coefficients, then we obtain smoothing polynomials of the form

$$S_k(x) = 1 + s_1 x + \ldots + s_k x^k \tag{12}$$

with $s_j = c_j(h\rho)^j$, where the c_j are constants. As a consequence, the non-zero coefficients of $x^j z^i$ in $R(x, z)$ are proportional to $(h\rho)^j$. This feature should be taken into account in the following accuracy considerations.

Accuracy conditions. The order of accuracy can be estimated by the local error $\varepsilon_{n+1} := y(t_{n+1}) - y_{n+1}$, where $y(t)$ denotes the exact solution through the point (t_n, y_n). In order to derive an expression for this error we first observe that, by virtue of the property that the components of $y(t)$ form a grid function defined on a grid with mesh size Δ, we may assume the existence of an integer p for which the grid functions

$$y_j(t) := \frac{D^j y(t)}{\Delta^{jp}}, \quad j = 0, 1, 2, \ldots$$

are bounded as Δ tends to 0 (p may be considered as the order of the difference matrix D). Furthermore, we use a notation by means of the forward shift operator E, i.e.,

$$Ey_j(t) = y_{j+1}(t)$$

We can now express the local error in the form

$$\begin{aligned}\varepsilon_{n+1}(\Delta, h) &= h[I - R_z(D, 0)]y'(t_n) + \tfrac{1}{2}h^2[I - R_{zz}(D, 0)]y''(t_n) + O(h^3)\\ &= h[1 - R_z(\Delta^p E, 0)]y'_0(t_n) + \tfrac{1}{2}h^2[1 - R_{zz}(\Delta^p E, 0)]y''_0(t_n) + O(h^3)\end{aligned} \tag{13}$$

The expansion (13) indicates that, for sufficiently smooth grid functions $y'(t_n)$ and $y''(t_n)$, the global error defined by $h^{-1}\varepsilon_{n+1}(\Delta, h)$ is controlled by the 'error' function

$$A_{mk}(\Delta^p, h) := 1 - R_z(\Delta^p, 0) + h[1 - R_{zz}(\Delta^p, 0)] + h^2 \tag{14}$$

Writing

$$A_{mk}(\Delta^p, h) = \sum_{ij} a_{ij} h^i \Delta^{jp} \tag{15a}$$

we obtain

$$\text{Global error} = \sum_{ij} |a_{ij}| O(h^i \Delta^{jp}) \tag{15b}$$

In the actual derivation of the error constants a_{ij} it is convenient to write the amplification polynomial in the form

$$P_m(x) = 1 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \ldots + \alpha_m x^m \tag{16}$$

so that the stability polynomial can be represented as

$$R(x, z) = 1 - z \sum_{j=1}^{m} \alpha_j [S_k(x)]^j [1 - \theta z]^{j-1} \tag{11'}$$

Here, the coefficients α_j are easily expressed in terms of the relaxation parameters. We can now express the error function $A_{mk}(\Delta^p, h)$ in terms of θ, the coefficients α_j and the smoothing polynomial S_k. It is easily verified that

$$R_z(x, 0) = -\sum_{j=1}^{m} \alpha_j [S_k(x)]^j, \quad R_{zz}(x, 0) = 2\theta \sum_{j=1}^{m} (j-1)\alpha_j [S_k(x)]^j$$

so that

$$A_{mk}(\Delta^p, h) = 1 + h + h^2 + \sum_{j=1}^{m} \alpha_j [1 - 2h\theta(j-1)][S_k(\Delta^p)]^j$$

The error constants a_{ij} can be determined by writing

$$A_{mk}(\Delta^p, h) = \sum_{j=0}^{mk} a_j(h)\Delta^{jp}, \quad a_{ij} := \frac{\mathrm{d}^i a_j(0)}{i!\,\mathrm{d}h^i} \tag{17}$$

The first three coefficient functions $a_j(h)$ are given by

$$a_0(h) = 1 + h + h^2 + \sum_{j=1}^{m} \alpha_j[1 - 2h\theta(j-1)]$$

$$a_1(h) = \sum_{j=1}^{m} j\alpha_j[1 - 2h\theta(j-1)]s_1$$

$$a_2(h) = \sum_{j=1}^{m} j\alpha_j[1 - 2h\theta(j-1)][s_2 + \tfrac{1}{2}(j-1)s_1^2]$$

(as in (12), the parameters s_j denote the coefficients of the smoothing polynomial $S_k(x)$).

We distinguish the case where the coefficient α_j is constant and s_j is of order h^j and the case where both α_j and s_j are constant. The corresponding values of the error constants a_{ij} are listed in Tables I and II.

We shall use the method parameters θ and α_j to make the first error constants a_{ij} and the smoothing parameters s_j vanish to maximize the stability boundary. In Table III the corresponding parameter values of θ and α_j, and the resulting relaxation parameters together with the orders of the global error are listed for $m = 1, 2, 3$. Since $\theta = 1/2$ for all m, we are in fact iterating the implicit midpoint rule. Furthermore, since the relaxation parameters all equal 1, the amplification polynomial $P_m(x)$ is identical with (10). Of course, this is a consequence of our decision to use all method parameters θ and α_j for increasing the order of accuracy. If one or more of the method parameters are used for improving the stability, then the amplification polynomial will not necessarily be equal to (10). Without claiming that it is the best strategy, we shall confine our considerations to methods with $\theta = 1/2$, $r^{(j)} = 1$ ($j = 1, \ldots, m$), and with smoothing polynomial

Table I. Error constants a_{ij} for α_j constant and $s_j = c_j(h\rho)^j$

	$j=0$	$j=1$	$j=2$
$i=0$	$1 + \sum_{j=1}^{m} \alpha_j$	0	0
$i=1$	$1 - 2\theta \sum_{j=1}^{m} (j-1)\alpha_j$	$c_1\rho \sum_{j=1}^{m} j\alpha_j$	0
$i=2$	1	$-2\theta c_1\rho \sum_{j=1}^{m} j(j-1)\alpha_j$	$\rho^2 \sum_{j=1}^{m} j\alpha_j[c_2 + \frac{1}{2}(j-1)c_1^2]$

Table II. Error constants a_{ij} for α_j and s_j constant

	$j=0$	$j=1$	$j=2$
$i=0$	$1 + \sum_{j=1}^{m} \alpha_j$	$s_1 \sum_{j=1}^{m} j\alpha_j$	$\sum_{j=1}^{m} j\alpha_j[s_2 + \frac{1}{2}(j-1)s_1^2]$
$i=1$	$1 - 2\theta \sum_{j=1}^{m} (j-1)\alpha_j$	$-2\theta s_1 \sum_{j=1}^{m} j(j-1)\alpha_j$	$-2\theta \sum_{j=1}^{m} j(j-1)\alpha_j[s_2 + \frac{1}{2}(j-1)s_1^2]$
$i=2$	1	0	0

Table III. Specification of the Method(m, S_k)

m	θ	α_1	α_2	α_3	$r^{(1)}$	$r^{(2)}$	$r^{(3)}$	Global error $s_j = c_j(h\rho)^j$	s_j constant
1	$\frac{1}{2}$	-1			1			$O(h + h\rho\Delta^p)$	$O(h + \Delta^p)$
2	$\frac{1}{2}$	-2	1		1	1		$O(h^2 + h^2\rho\Delta^p)$	$O(h^2 + h\Delta^p + \Delta^{2p})$
3	$\frac{1}{2}$	-3	3	-1	1	1	1	$O(h^2 + h^2\rho^2\Delta^{2p})$	$O(h^2 + \Delta^{2p})$

S_k such that the stability is optimal in some sense. These methods will be denoted by Method(mS_k).

Stability conditions. According to Table III and using (10) and (11), the stability polynomial of Method(m, S_k) reduces to the form

$$R(x, z) := \frac{1 + [\frac{1}{2} - (1 - S_k(x)[1 - \frac{1}{2}z])^m]z}{1 - \frac{1}{2}z} \tag{18}$$

The region in the (x, z)-space where the modulus of this polynomial is bounded by 1 will be called the *stability region.*

Example 1. Consider Method(1, S_1) where $S_1(x) = 1 + s_1 x$, s_1 being a positive constant. Then the stability polynomial (18) is given by

$$R(x, z) = 1 + z(1 + s_1 x)$$

In the (x, z)-plane, the stability region is given by $|1 + z(1 + s_1 x)| \leqslant 1$. Let us consider this region for imaginary values of x and z. In Example 2 below, it will be shown that the largest possible imaginary stability boundary equals 1 and is obtained for $s_1 = 1$. Setting $x = \mathrm{i}\xi$, $z = \mathrm{i}h\rho\zeta$ with $h\rho = 1$ and $-1 \leqslant \xi, \zeta \leqslant 1$, we find that, in the (ξ, ζ)-plane, the stability region is bounded by the lines $\zeta = 0$ and $\zeta\xi^2 - 2\xi + \zeta = 0$. □

In order to obtain manageable stability conditions, we consider the model situation where the difference matrix D actually equals the normalized Jacobian $(h\rho)^{-1}Z$, so that $S = S_k((h\rho)^{-1}Z)$. Evidently, for a given value of $h\rho$, the iterated θ-method is stable if the modulus of the stability polynomial $R((h\rho)^{-1}z, z)$ is bounded by 1 when z runs through the eigenvalues of the matrix Z. Such a value of $h\rho$ will be called a *stable hρ-value.* In actual computation, where the value is estimated during the integration process, it is recommended to require that there is a sufficiently large interval of stable $h\rho$-values. To be more precise, suppose that the method has the interval $[\alpha, \beta]$ as its range of stable $h\rho$-values, and let ρ^* be an estimate for the true spectral radius ρ. Usually, we desire the largest possible step h, so that we set $h = \beta/\rho^*$. Since it is required that $h\rho \in [\alpha, \beta]$, we have only stability if ρ^* satisfies the inequality $\rho \leqslant \rho^* \leqslant \beta\rho/\alpha$.

In this paper, we shall assume that the spectrum of the matrix Z is (essentially) imaginary. Thus, we are faced with the problem to keep the values

$$R((h\rho)^{-1}z, z) := \frac{1 + [\frac{1}{2} - (1 - S_k((h\rho)^{-1}z)[1 - \frac{1}{2}z])^m]z}{1 - \frac{1}{2}z}, \quad z \in [-\mathrm{i}h\rho, \mathrm{i}h\rho] \tag{19}$$

on the unit disk for a maximum range of $h\rho$-values by a judicious choice of the smoothing polynomial S_k. In the ideal case where the whole interval $[0, \beta]$ contains stable $h\rho$-values, β is called the *imaginary stability boundary* and will be denoted by β_{imag}.

Example 2. Consider again Method$(1, S_1)$ of Example 1. It is easily verified that the polynomial (19), i.e.

$$R((h\rho)^{-1}z, z) = 1 + z(1 + s_1(h\rho)^{-1}z)$$

assumes values on the unit disk for all values of z in the imaginary interval

$$[-\mathrm{i}H, \mathrm{i}H], \quad H := \sqrt{\frac{h\rho(2s_1 - h\rho)}{s_1^2}}$$

provided that $h\rho \leqslant 2s_1$. Thus, if z runs through the interval $[-\mathrm{i}h\rho, \mathrm{i}h\rho]$, then we should require $H \geqslant h\rho$ resulting in $h\rho \leqslant 2s_1/(1 + s_1^2)$. This leads us to put $s_1 = 1$ to obtain the largest possible imaginary stability boundary $\beta_{\text{imag}} = 1$. □

More generally, we have the following theorem on the maximal attainable imaginary stability boundaries of the iterated implicit midpoint rule.

Theorem 1. The imaginary stability boundary of Method(m, S_k) can never exceed $m(k + 1) - 1$

Proof. The polynomial $R((h\rho)^{-1}z, z)$ is of the form $1 + z + \beta_2 z^2 + \beta_3 z^3 + \ldots + \beta_{m(k+1)} z^{m(k+1)}$. It is known[12] that the imaginary stability boundary of such polynomials cannot exceed the degree of the polynomial minus 1, i.e. $\beta_{\text{imag}} \leqslant m(k + 1) - 1$. □

In the following sections we consider one-stage, two-stage and three-stage methods in which the smoothing polynomial is determined such that the stability polynomial (19) is a polynomial with fixed coefficients possessing a large imaginary stability boundary β_{imag}. As we already observed, the coefficients of the polynomials S_k obtained, and therefore the generated methods Method(m, S_k), are $h\rho$-dependent. Since it is sometimes convenient to have methods independent of ρ, we also consider methods where $h\rho$ is replaced by β_{imag}, so that the coefficients of the smoothing polynomial are constant. We remark that, as a consequence, the range of stable $h\rho$-values may change.

ONE-STAGE METHODS

In this section we consider the Method$(1, S_k)$ defined in Table III. The stability polynomial of these methods is given by

$$R(x, z) = 1 + zS_k(x) \tag{20}$$

From this expression we immediately conclude:

Theorem 2. Method$(1, S_k)$ *has a zero imaginary stability boundary if S_k is real-valued.* □

However, if the smoothing polynomial is complex-valued, then methods with non-zero imaginary stability boundaries are easily constructed.

Methods of $O(h + h\rho\Delta^p)$

Our starting point is a result of Kinnmark and Gray[5] stating that the polynomials $I_{k+1}(z)$ which satisfy $I_{k+1}(0) = I'_{k+1}(0) = 1$ and which assume values on the unit disk in the largest possible imaginary interval $[-ik, ik]$, are given by

$$I_{k+1}(z) := i^k \left\{ T_k\left(\frac{z}{ik}\right) + \frac{1}{2} i \left[T_{k+1}\left(\frac{z}{ik}\right) - T_{k-1}\left(\frac{z}{ik}\right) \right] \right\} \tag{21}$$

Thus, by identifying $R((h\rho)^{-1}z, z)$ with $I_{k+1}(z)$, that is,

$$S_k(x) = \frac{I_{k+1}(h\rho x) - 1}{h\rho x} \tag{22}$$

we achieve that (in the model situation) the range of stable $h\rho$-values is given by $0 \leqslant h\rho \leqslant k$, so that the imaginary stability boundary β_{imag} equals k. Notice that, according to Theorem 1, this value is optimal. It follows from Table III that the method is $O(h + h\rho\Delta^p)$. For future reference, we list the first three smoothing polynomials (Table IV).

Methods of $O(h + \Delta^p)$

Let us define

$$S_k(x) = \frac{I_{k+1}(kx) - 1}{kx} \tag{22*}$$

so that in the model situation where $x = z/h\rho$ we have

$$R((h\rho)^{-1}z, z) = 1 + \frac{h\rho}{k}\left[I_{k+1}\left(\frac{kz}{h\rho}\right) - 1 \right] \tag{23}$$

The first three smoothing polynomials together with the ranges of stable $h\rho$-values of this stability polynomial are given in Table V. It turns out that here the stability range equals $[0, k]$ so that at least for $k \leqslant 3$ the 'fixed smoothing polynomial' versions of Method(1, S_k) possess a non-zero imaginary stability boundary $\beta_{\text{imag}} = k$. We did not succeed in proving this property for all k. Finally, we remark that the methods are $O(h + \Delta^p)$ accurate for all values of k (cf. Table II).

Table IV. Smoothing polynomials for use in Method(1, S_k)

k	$S_k(x)$	β_{imag}
1	$1 + h\rho x$	1
2	$1 + \frac{1}{2}h\rho x + \frac{1}{4}(h\rho x)^2$	2
3	$1 + \frac{5}{9}h\rho x + \frac{4}{27}(h\rho x)^2 + \frac{4}{81}(h\rho x)^3$	3

Table V. Smoothing polynomials for use in Method(1, S_k)

k	$S_k(x)$	Stable $h\rho$-range
1	$1 + x$	[0, 1]
2	$1 + x + x^2$	[0, 2]
3	$\frac{1}{3}(3 + 5x + 4x^2 + 4x^3)$	[0, 3]

TWO-STAGE METHODS

Next we consider the Method(2, S_k) as defined in Table III. The stability polynomial of these methods is given by

$$R(x, z) = 1 + zS_k(x)\{2 - S_k(x)[1 - \tfrac{1}{2}z]\} \tag{24}$$

For this polynomial we have

Theorem 3.

(a) If the eigenvalues of Z are purely imaginary, and if the smoothing polynomial is real-valued, then the stability condition of Method (2, S_k) *is given by*

$$S_-(\zeta) \leqslant S_k(x) \leqslant S_+(\zeta), \quad |\zeta| \leqslant h\rho, \quad -1 \leqslant x \leqslant 0 \tag{25}$$

where

$$S_\pm(\zeta) := \frac{4 \pm \sqrt{4 - 3\zeta^2}}{2 + \frac{1}{2}\zeta^2}$$

(b) If the conditions of (a) are satisfied, then the imaginary stability boundary cannot exceed the value $\sqrt{4/3}$.

Proof.

(*a*) If $S_k(x)$ is real-valued and if $z = \mathrm{i}\zeta$ with ζ real, then we have

$$|R(x, z)|^2 = [1 - \tfrac{1}{2}\zeta^2(S_k(x))^2]^2 + \zeta^2[S_k(x)(2 - S_k(x))]^2 \tag{26}$$

The stability condition $|R(x, z)| \leqslant 1$ leads to the inequality

$$[1 + \tfrac{1}{4}\zeta^2][S_k(x)]^2 - 4S_k(x) + 3 \leqslant 0 \tag{27}$$

for all non-zero values of $S_k(x)$. This leads straightforwardly to the condition (25).

(*b*) For real values of $S_k(x)$ condition (27) can be satisfied only if ζ^2 satisfies the condition $4 \geqslant 3\zeta^2$, that is

$$|\zeta| \leqslant \sqrt{\tfrac{4}{3}}$$

Thus, the imaginary stability boundary can never exceed the value $\sqrt{4/3}$. □

This theorem reveals that *real-valued* smoothing polynomials are not very effective in hyperbolic schemes. Therefore, we have concentrated on more general *complex-valued* smoothing polynomials in order to increase the imaginary stability boundary. However, since the coefficients β_j of $R((h\rho)^{-1}z, z)$ are not free but functions of the k smoothing coefficients s_j, we should not expect to find boundaries as large as the upper bound $m(k + 1) - 1$ stated in Theorem 1.

Methods of $O(h^2 + h^2\rho\Delta^p)$

The Method(2, S_k) employing the smoothing polynomial $S_k(x) = 1 + s_1x + \ldots + s_kx^k$ contains the k free parameters $\{s_1, \ldots, s_k\}$ for maximizing the imaginary stability boundary β_{imag}. In the model situation the stability polynomial is of the form

$$R((h\rho)^{-1}z, z) = 1 + zS_k((h\rho)^{-1}z)\{2 - S_k((h\rho)^{-1}z)[1 - \tfrac{1}{2}z]\} \tag{24'}$$

We used a numerical search in order to determine suitable parameter values. In Table VI we list a

Table VI. Smoothing polynomials for use in Method(2, S_k)

k	$S_k(x)$	β_{imag}
1	$1 + \frac{1}{4}h\rho x$	2·5
2	$1 + \frac{11}{50}h\rho x + \frac{1}{25}(h\rho x)^2$	3·75
3	$1 + \frac{7}{25}h\rho x + \frac{3}{100}(h\rho x)^2 + \frac{3}{400}(h\rho x)^3$	6

few smoothing polynomials and the generated imaginary stability boundaries which are appropriate for use in Method(2, S_k). These boundaries are about 83, 75 and 85 per cent of the upper bounds given in Theorem 1. According to Table III the corresponding methods are of $O(h^2 + h^2\rho\Delta^p)$.

Methods of $O(h^2 + h\Delta^p + \Delta^{2p})$

By replacing in Table VI the problem parameter $h\rho$ by β_{imag} we obtain smoothing polynomials with constant coefficients generating methods of $O(h^2 + h\Delta^p + \Delta^{2p})$. In Table VII, the analogue of Table VI is given. Unlike the case of one-stage methods, the price for having fixed smoothing polynomials is a reduced interval of stable $h\rho$ values. However, we shall see that in actual computation the intervals of unstable $h\rho$ values hardly influence the accuracy. Notice that the $h\rho$-independent version of Method(2, S_3) allows larger steps than the $h\rho$-dependent version.

THREE-STAGE METHODS

Finally, we consider the Method(3, S_k) of Table III.

Methods of $O(h^2 + h^2\rho^2\Delta^{2p})$

Similar to the previous section we computed smoothing polynomials for use in the $O(h^2 + h^2\rho^2\Delta^{2p})$ version of Method(3, S_k). See Table VIII.

Methods of $O(h^2 + \Delta^{2p})$

The analogue of Table VII is given by Table IX. Again the method using the third degree smoothing polynomial is rather sensitive to an accurate estimate of the spectral radius.

Table VII. Smoothing polynomials for use in Method(2, S_k)

k	$S_k(x)$	Stable $h\rho$-range
1	$\frac{1}{8}(8 + 5x)$	[1·25, 2·5]
2	$\frac{1}{80}(80 + 66x + 45x^2)$	[0, 0·89] + [2·89, 3·75]
3	$\frac{1}{50}(50 + 84x + 54x^2 + 81x^3)$	[0, 0·94] + [4·62, 4·67] + [4·85, 5·02] + [5·13, 5·42] + [5·47, 6·25]

Table VIII. Smoothing polynomials for use in Method(3, S_k)

k	$S_k(x)$	β_{imag}
1	$1 + \frac{1}{8}h\rho x$	2·6
2	$1 + \frac{3}{40}h\rho x + \frac{3}{125}(h\rho x)^2$	5·5
3	$1 + \frac{367}{2000}h\rho x + \frac{51}{2000}(h\rho x)^2 + \frac{1}{250}(h\rho x)^3$	5·75

Table IX. Smoothing polynomials for use in Method(3, S_k)

k	$S_k(x)$	Stable $h\rho$-range
1	$\frac{1}{40}(40 + 13x)$	[0·58, 0·62] + [1·08, 2·6]
2	$\frac{1}{2000}(2000 + 825x + 1452x^2)$	[0·63, 0·84] + [3·47, 5·54]
3	$\frac{1}{32\,000}(32\,000 + 33\,764x + 26\,979x^2 + 24\,334x^3)$	[5·61, 5·75]

SMOOTHING MATRICES

In our numerical experiments we integrated the semidiscrete hyperbolic problem given in (2). However, in order to obtain a Jacobian matrix with an appropriate difference structure, we modify this system slightly. Instead of substituting the boundary value y_0 in the equation for y_1, we convert the boundary condition $y_0 = b(t)$ into a differential equation by analytical or numerical differentiation. Thus, if $b'(t)$ is available, then we add to (2) the equation

$$\frac{\mathrm{d}y_0}{\mathrm{d}t} = b'(t) \tag{28}$$

The normalized Jacobian matrix of the right-hand-side function of the system $\{(2), (28)\}$ is approximated by

$$D = \frac{1}{2}\begin{pmatrix} 0 & & & \cdots & & & 0 \\ 1 & 0 & -1 & & & & \\ 0 & 1 & 0 & -1 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & 1 & 0 & -1 & 0 \\ & & & & 1 & 0 & -1 \\ 0 & & & \cdots & -1 & 4 & -3 \end{pmatrix} \tag{29}$$

We shall use this difference matrix for generating the smoothing matrices $S = S_k(D)$ when integrating problems of the form $\{(2), (28)\}$. Notice that (29) does have a difference structure indeed, which would not be the case if y_0 is eliminated from the equation for y_1.

For an efficient implementation it is desirable to compute S in advance. In the Appendix we have listed the matrices $S = S_{mk}$ associated with Method(m, S_k) for all polynomials S_k specified in Tables V, VII and IX.

NUMERICAL EXAMPLES

In our experiments, we chose problems of the form $\{(2), (28)\}$ and we discretized the problems as indicated in the Introduction. The initial and boundary conditions were taken from the exact solution $u(x, t)$. Thus, by specifying the functions $u(x, t)$ and $a(x, t, u)$, and the grid size Δx, the initial-value problem $\{(2), (28)\}$ is completely defined.

We represent the maximum absolute error (with respect to the solution u) at the end point of the integration interval in the form 10^{-sd}, where sd may be considered as the number of correct significant digits. In the sections below we present sd/sd^*-values for a few problems. Here, sd and sd^* respectively correspond to the $h\rho$-dependent and $h\rho$-independent versions of the smoothing polynomial occurring in Method(m, S_k). To provide a reference, we also list the sd-value obtained for the implicit midpoint rule, when solved exactly using Newton iteration. Unstable results are indicated by an asterisk.

Model problem

Table X lists results for a model problem semidiscretized on a fixed grid with various values of the time step h. From this table the following conclusions can be drawn.

1. All methods are stable when $h\rho$ lies in the range given in the Tables IV–IX.
2. Except for Method(2, S_1) the accuracy is not affected by the intervals of instability associated with the $h\rho$-independent versions of the two- and three-stage methods.
3. The first-order and zero-order time discretization error of the $h\rho$-dependent and $h\rho$-independent versions of the one-stage methods is clearly recognizable (in the two- and three-stage methods, the time discretization error is hidden by the space and smoothing errors).
4. Except for the $h\rho$-independent version of Method(2, S_k) the accuracy is not affected by the degree k of the smoothing polynomial.
5. The $h\rho$-independent and $h\rho$-dependent versions of the three-stage method as well as the $h\rho$-dependent version of the two-stage method produce results of the same accuracy as the implicit midpoint rule; however, this rule requires much more computational effort.

Linear problem with varying coefficients

Our second problem differs from the model problem by an (x, t)-dependent coefficient function a. From the various methods we selected the $(m, k) = (1, 3), (2, 3)$ and $(3, 2)$ methods which possess the best stability characteristics. Together with the Newton iterated implicit midpoint rule the sd-values obtained are listed in a box like

Method(1, S_3)	Method(2, S_3)
Method(3, S_2)	Newton

Table XI presents the sd-values obtained. Again we observe the correct order behaviour of the various methods and, similar to the model example, the intervals of instability associated with the $h\rho$-independent version did not manifest themselves. A comparison of the accuracy behaviour of the $h\rho$-dependent and $h\rho$-independent versions reveals that in the three-stage scheme both versions yield the accuracy of the implicit midpoint rule. In the one- and two-stage methods, however, we observe a difference in favour of the $h\rho$-dependent version.

Table X. sd/sd^*-values obtained for $a := -1$, $u = \sin(t - x)$, $0 \leqslant t \leqslant 1$ and $\Delta x = 1/80$

	$m = 1$			$m = 2$			$m = 3$			
h^{-1}	$k = 1$	$k = 2$	$k = 3$	$k = 1$	$k = 2$	$k = 3$	$k = 1$	$k = 2$	$k = 3$	Newton
10	*/*	*/*	*/*	*/*	*/*	*/0·9	*/*	*/*	*/*	3·1
20	*/*	*/*	*/2·0	*/*	2·1/3·6	3·6/3·7	*/*	3·6/3·6	3·6/3·6	3·7
40	*/*	2·2/2·2	2·2/2·2	4·2/4·2	4·3/3·5	4·2/3·8	4·1/4·1	4·1/4·1	4·1/4·1	4·1
80	2·5/2·5	2·5/2·5	2·5/2·5	4·5/4·4	4·6/4·4	4·5/3·6	4·4/4·4	4·4/4·4	4·4/4·4	4·4
160	2·8/2·7	2·8/2·7	2·8/2·4	4·6/3·7	4·6/4·2	4·6/3·5	4·5/4·5	4·5/4·5	4·5/4·6	4·5
320	3·1/2·6	3·1/2·6	3·1/2·4	4·6/4·3	4·6/4·2	4·6/3·5	4·6/4·6	4·6/4·6	4·6/4·5	4·6
640	3·4/2·6	3·4/2·6	3·4/2·4	4·6/4·3	4·6/4·1	4·6/3·5	4·6/4·6	4·6/4·6	4·6/4·5	4·6

Table XI. sd/sd^*-values obtained for $a: = -x/(2(1 + t))$, $u = \sin(x^2(1 + t)^{-1})$ and $0 \leqslant t \leqslant 1$

	$\Delta x = \frac{1}{20}$		$\Delta x = \frac{1}{40}$		$\Delta x = \frac{1}{80}$		$\Delta x = \frac{1}{160}$		$\Delta x = \frac{1}{320}$	
$h = \frac{1}{5}$	1·8/1·7	2·8/2·3	1·8/1·8	2·9/2·9	*/1·5	2·9/2·9	*/1·4	*/1·4	*/1·4	*/*
	2·9/3·0	3·0	3·0/3·0	3·1	3·1/3·0	3·2	*/*	3·2	*/*	3·2
$h = \frac{1}{10}$	2·1/1·5	3·2/2·3	2·1/2·0	3·4/2·9	2·1/2·1	3·5/3·5	*/1·8	3·6/3·5	*/0·8	*/*
	3·2/3·3	3·3	3·5/3·6	3·6	3·7/3·7	3·8	3·8/3·7	3·8	*/*	3·8
$h = \frac{1}{20}$	2·4/1·4	3·3/2·2	2·4/1·8	3·7/2·8	2·4/2·3	4·0/3·4	2·4/2·4	4·1/4·0	*/1·8	3·6/4·1
	3·4/3·3	3·4	3·8/3·8	3·8	4·2/4·2	4·2	4·3/4·3	4·4	4·4/4·4	4·4
$h = \frac{1}{40}$	2·8/1·4	3·4/2·2	2·7/1·7	3·9/2·8	2·7/2·1	4·3/3·4	2·7/2·6	4·6/4·0	2·7/2·7	4·7/4·6
	3·4/3·4	3·4	3·9/3·9	3·9	4·4/4·4	4·4	4·8/4·8	4·8	4·9/5·0	5·0
$h = \frac{1}{80}$	3·2/1·4	3·4/2·2	3·0/1·7	3·9/2·8	3·0/2·0	4·5/3·4	3·0/2·4	4·9/4·0	3·0/2·9	5·2/4·6
	3·4/3·4	3·4	3·9/3·9	3·9	4·5/4·5	4·5	5·0/5·0	5·0	5·4/5·4	5·4

Table XII. *sd*/*sd**-values obtained for $a := -u$, $u = \frac{1}{2}(-t + \sqrt{t^2 + 4x})$ and $1 \leqslant t \leqslant 2$

	$\Delta x = \frac{1}{20}$		$\Delta x = \frac{1}{40}$		$\Delta x = \frac{1}{80}$		$\Delta x = \frac{1}{160}$		$\Delta x = \frac{1}{320}$	
$h = \frac{1}{10}$	2·2/1·7 4·1/3·1	3·5/2·1 4·1	2·1/2·0 4·3/3·4	3·1/2·3 4·3	*/2·2 4·3/3·7	2·8/2·6 4·4	*/2·3 */0·5	*/2·9 4·4	*/0·5 */*	*/* 4·4
$h = \frac{1}{20}$	2·5/1·7 4·3/3·1	4·0/2·1 4·3	2·4/1·9 4·7/3·4	3·7/2·3 4·7	2·4/2·2 4·9/3·7	3·3/2·6 4·9	*/2·5 4·6/4·0	3·1/2·9 5·0	*/* */*	*/* 5·0
$h = \frac{1}{40}$	2·8/1·7 4·4/3·1	4·3/2·1 4·4	2·7/1·9 4·9/3·3	4·3/2·3 4·9	2·7/2·2 5·3/3·6	3·9/2·6 5·3	2·7/2·5 5·5/3·9	3·6/2·9 5·5	*/2·8 4·8/4·2	3·4/3·2 5·6
$h = \frac{1}{80}$	3·1/1·7 4·4/3·1	4·4/2·1 4·4	3·0/1·9 5·0/3·3	4·9/2·3 5·0	3·0/2·2 5·5/3·6	4·5/2·6 5·5	3·0/2·5 5·9/3·9	4·1/2·9 5·9	2·9/2·8 6·1/4·2	3·8/3·2 6·1
$h = \frac{1}{160}$	3·4/1·7 4·4/3·0	4·4/2·1 4·4	3·3/1·9 5·0/3·3	5·0/2·3 5·0	3·3/2·2 5·6/3·6	5·1/2·6 5·6	3·3/2·5 6·1/3·9	4·7/2·9 6·1	3·2/2·8 6·5/4·2	4·4/3·2 6·5

Non-linear problem

Our third problem is non-linear, that is, the coefficient function a depends on u. The analogue of Table XI is given by Table XII. In this example, the difference in the accuracies produced by the $h\rho$-dependent and $h\rho$-independent versions is much larger than in the previous examples, especially for the two- and three-stage methods. Apparently, in this non-linear problem, the influence of the smoothing error on the global error is rather large, resulting in a superior behaviour of the $h\rho$-dependent version of the method, particularly for small h. However, we should realize that in practical applications usually the largest possible step size will be used. For these values of h the difference of both versions is much less pronounced.

Furthermore, we observe that the three-stage method using the $h\rho$-dependent version produces (for all stable step sizes) almost the same accuracy as the implicit midpoint rule. However, this Method(3, S_2) requires considerably less computational effort than the implicit midpoint rule (in our implementation we measured a factor 8).

APPENDIX

For the actual computer solution to the semidiscretization $\{(2), (28)\}$ of the initial-boundary-value problem (1) by means of the $h\rho$-independent methods, the following explicitly written out algorithm may be used,

$$y^{(0)} = y_n; \quad y^{(j)} = y^{(j-1)} - S_{mk}\left[y^{(j-1)} - y_n - hf\left(\frac{t_n + t^{(j-1)}}{2}, \frac{y_n + y^{(j-1)}}{2}\right)\right]$$

$$t^{(0)} = t_n; \quad t^{(j)} = t_n + h$$

where $j = 1, \ldots, m$ and where S_{mk} is the smoothing matrix which is defined by one of the following matrices (notice that the dimension of these smoothing matrices equals the number of semidiscrete differential equations in $\{(2), (28)\}$):

$$S_{11} = \frac{1}{2}\begin{pmatrix} 2 & 0 & & & & & \\ 1 & 2 & -1 & 0 & & & \\ 0 & 1 & 2 & -1 & 0 & & \\ & & & \cdots & & & \\ & & 0 & 1 & 2 & -1 & 0 \\ & & & 0 & 1 & 2 & -1 \\ & & & 0 & -1 & 4 & -1 \end{pmatrix}$$

$$S_{12} = \frac{1}{4}\begin{pmatrix} 4 & 0 & & & & & & & \\ 2 & 3 & -2 & 1 & 0 & & & & \\ 1 & 2 & 2 & -2 & 1 & 0 & & & \\ & & & & \cdots & & & & \\ & & & 0 & 1 & 2 & 2 & -2 & 1 \\ & & & & 0 & 1 & 3 & -1 & 1 \\ & & & & 0 & -1 & 5 & -3 & 3 \end{pmatrix}$$

$$S_{13} = \frac{1}{6}\begin{pmatrix} 6 & 0 & & & & & & & \\ 4 & 4 & -3 & 2 & -1 & 0 & & & \\ 2 & 3 & 2 & -2 & 2 & -1 & 0 & & \\ 1 & 2 & 2 & 2 & -2 & 2 & -1 & 0 & \\ & & & \cdots & & & & & \\ & 0 & 1 & 2 & 2 & 2 & -2 & 2 & -1 \\ & & 0 & 1 & 2 & 2 & 1 & 1 & -1 \\ & & & 0 & 1 & 3 & -2 & 7 & -3 \\ & & & 0 & -1 & 5 & -6 & 11 & -3 \end{pmatrix}$$

$$S_{21} = \frac{1}{16}\begin{pmatrix} 16 & 0 & & & & & & \\ 5 & 16 & -5 & 0 & & & & \\ 0 & 5 & 16 & -5 & 0 & & & \\ & & & \cdots & & & & \\ & & & 0 & 5 & 16 & -5 & 0 \\ & & & 0 & 0 & 5 & 16 & -5 \\ & & & & 0 & -5 & 20 & 1 \end{pmatrix}$$

$$S_{22} = \frac{1}{320}\begin{pmatrix} 320 & 0 & & & & & & \\ 132 & 275 & -132 & 45 & 0 & & & \\ 45 & 132 & 230 & -132 & 45 & 0 & & \\ 0 & 45 & 132 & 230 & -132 & 45 & 0 & \\ & & & & \cdots & & & \\ & 0 & 45 & 132 & 230 & -132 & 45 & 0 \\ & & 0 & 45 & 132 & 230 & -132 & 45 \\ & & & 0 & 45 & 177 & 95 & 3 \\ & & & & -45 & 183 & 33 & 149 \end{pmatrix}$$

$$
S_{23} = \frac{1}{400}\begin{pmatrix}
400 & 0 & & & & & & & & \\
255 & 292 & -174 & 108 & -81 & 0 & & & & \\
108 & 174 & 184 & -93 & 108 & -81 & 0 & & & \\
81 & 108 & 93 & 184 & -93 & 108 & -81 & 0 & & \\
0 & 81 & 108 & 93 & 184 & -93 & 108 & -81 & 0 & \\
 & & & & \cdots & & & & & \\
 & 0 & 81 & 108 & 93 & 184 & -93 & 108 & -81 & 0 \\
 & & 0 & 81 & 108 & 93 & 184 & -93 & 108 & -81 \\
 & & & 0 & 81 & 108 & 93 & 103 & 150 & -135 \\
 & & & & 0 & 81 & 189 & -285 & 751 & -336 \\
 & & & & & -81 & 459 & -795 & 1209 & -392
\end{pmatrix}
$$

$$
S_{31} = \frac{1}{80}\begin{pmatrix}
80 & 0 & & & & \\
13 & 80 & -13 & 0 & & \\
0 & 13 & 80 & -13 & 0 & \\
 & & \cdots & & & \\
 & 0 & 13 & 80 & -13 & 0 \\
 & & 0 & 13 & 80 & -13 \\
 & & 0 & -13 & 52 & 41
\end{pmatrix}
$$

$$
S_{32} = \frac{1}{4000}\begin{pmatrix}
4000 & 0 & & & & & & \\
825 & 3274 & -825 & 726 & 0 & & & \\
726 & 825 & 2548 & -825 & 726 & 0 & & \\
0 & 726 & 825 & 2548 & -825 & 726 & 0 & \\
 & & & & \cdots & & & \\
 & 0 & 726 & 825 & 2548 & -825 & 726 & 0 \\
 & & 0 & 726 & 825 & 2548 & -825 & 726 \\
 & & & 0 & 726 & 1551 & 370 & 1353 \\
 & & & & -726 & 4257 & -4686 & 5155
\end{pmatrix}
$$

$$
S_{33} = \frac{1}{128\,000}\left(\begin{array}{cccccccccc}
128\,000 & 0 & & & & & & & & \\
55\,361 & 101\,021 & -43\,194 & 26\,979 & -12\,167 & 0 & & & & \\
26\,979 & 43\,194 & 74\,042 & -31\,027 & 26\,979 & -12\,167 & 0 & & & \\
12\,167 & 26\,979 & 31\,027 & 74\,042 & -31\,027 & 26\,979 & -12\,167 & 0 & & \\
0 & 12\,167 & 26\,979 & 31\,027 & 74\,042 & -31\,027 & 26\,979 & -12\,167 & 0 & \\
 & & & & \cdots & & & & & \\
 & 0 & 12\,167 & 26\,979 & 31\,027 & 74\,042 & -31\,027 & 26\,979 & -12\,167 & 0 \\
 & & 0 & 12\,167 & 26\,979 & 31\,027 & 74\,042 & -31\,027 & 26\,979 & -12\,167 \\
 & & & 0 & 12\,167 & 26\,979 & 31\,027 & 61\,875 & 5474 & -9522 \\
 & & & & 0 & 12\,167 & 39\,146 & -14\,996 & 126\,942 & -35\,259 \\
 & & & & & -12\,167 & 58\,190 & -61\,180 & 131\,514 & 11\,643
\end{array}\right)
$$

If these matrices are used, then the maximally stable time step is given by $h = \beta_{mk}\rho^{-1}$, where ρ denotes the spectral radius of $\partial f/\partial y$ and β_{mk} is given in the following table:

β_{mk}	$k = 1$	$k = 2$	$k = 3$
$m = 1$	1	2	3
$m = 2$	2·5	3·75	6·25
$m = 3$	2·6	5·54	5·75

If smaller steps are used, then weak unstable behaviour may occur for $m \geqslant 2$ (cf. Tables VII* and IX*).

REFERENCES

1. E. Hairer, S. P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems*, Springer-Verlag, Berlin, 1987.
2. P. J. van der Houwen and B. P. Sommeijer, 'Improving the stability of predictor–corrector methods by residue smoothing', *JIMA*, to appear.
3. P. J. van der Houwen and B. P. Sommeijer, 'Smoothed predictor–corrector methods for solving partial differential equations', *Proc. Int. Conf. on Numerical Mathematics*, Singapore 1988, ISNM **86**, Birkhäuser, Basel, 1989, pp. 201–224.
4. A. Jameson, 'The evolution of computational methods in aerodynamics', *J. Appl. Mech.*, **50**, 1052–1076 (1983).
5. I. P. E. Kinnmark and W. G. Gray, 'One step integration methods with maximum stability regions', *Math. Comp. Simulation*, **XXVI**, 87–92 (1984).
6. I. P. E. Kinnmark and W. G. Gray, 'One step integration methods of third-fourth order accuracy with large hyperbolic stability limits', *Math. Comp. Simulation*, **XXVI**, 181–188 (1984).
7. I. P. E. Kinnmark and W. G. Gray, 'Fourth-order accurate one-step integration methods with large imaginary stability limits', *Numer. Methods Partial Differential Equations*, **2**, 63–70 (1986).
8. A. Lerat, 'Une class de schémas aux différence implicites pour les systemes hyperboliques de lois de conservation', *C.R. Acad. Sci. (Paris)*, **288A**, 1033–1036 (1979).
9. A. R. Mitchell and D. F. Griffiths, *The Finite Difference Method in Partial Differential Equations*, Wiley, Chichester, 1980.
10. E. Turkel, 'Acceleration to a steady state for the Euler equations', in *Numerical Methods for the Euler Equations of Fluid Dynamics*, SIAM Publications, Philadelphia, 1985, pp. 218–311.
11. R. S. Varga, 'A comparison of the successive over-relaxation method and semi-iterative methods using Chebyshev polynomials', *SIAM J. Appl. Math.*, **5**, 39–47 (1957).
12. R. Vichnevetsky, 'New stability theorems concerning one-step integration methods for ordinary differential equations', *Math. Comp. Simulation*, **XXV**, 199–205 (1983).