STABILIZATION OF EXPLICIT DIFFERENCE SCHEMES BY SMOOTHING TECHNIQUES

P.J. van der Houwen
Amsterdam, The Netherlands

1. Introduction

Smoothing techniques are frequently applied in the numerical solution of (initial-)boundary-value problems for partial differential equations. Usually, the smoothing procedure consists of applying a matrix S to certain grid functions appearing in the difference scheme approximating the (initial-)boundary-value problem. The aim is the reduction of the magnitude of high frequency components occurring in the Fourier expansion of these grid functions without affecting the lower frequency components too much. These high frequencies are often unwanted components which enter during the numerical calculations. A simple example of a smoothing matrix is the 'averaging' matrix S defined by $\mathbf{g} = S\mathbf{f}$, where

$$(1) \qquad \begin{aligned} g_1 &= f_1, \\ g_j &= [\, f_{j-1} + f_{j+1} \,] / 2, \quad j = 2, \ldots, m-1, \\ g_m &= f_m, \end{aligned}$$

with f_j and g_j, $j = 1, \ldots, m$, denoting the components of $\mathbf{f}$ and $\mathbf{g}$, respectively.

We shall give a few examples where smoothing has successfully been applied.

1.1. Smoothing of the numerical solution

The most common situation is the case where the numerical solution itself is smoothed. We mention the work of Shuman [10] of 1957 who used smoothing matrices of type (1) in combination with hyperbolic difference schemes. After each integration step, the high frequencies introduced by the difference scheme were removed from the numerical solution just by premultiplying the solution vector by the matrix S. The matrix S is sometimes called a Shuman filter.

A more sophisticated example is the Richtmyer scheme for solving the initial-value problem for the hyperbolic equation

$$(2) \qquad \partial u(t,x) / \partial t = L(u(t,x)),$$

where L is a hyperbolic differential operator with respect to the (one-dimensional) space variable x. Richtmyer's scheme can be derived as follows. Let the system of ordinary differential equations

$$(3) \qquad dy(t) / dt = \mathbf{f}(t, y(t))$$

represent a semi-discretization for (2); here, the vector $\mathbf{y}(t)$ denotes an approximation to $u(t,x)$ on some grid on the x-axis. By applying the two-stage, explicit, second-order Runge-Kutta method of Runge (1895)

$$\begin{array}{c|cc} 0 & 0 & \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array},$$

we obtain the difference scheme

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t\, \mathbf{f}(t_n + \Delta t/2,\ \mathbf{y}_n + \Delta t\, \mathbf{f}(t_n, \mathbf{y}_n)/2). \tag{4}$$

It is well known that this scheme is unconditionally unstable when applied to hyperbolic equations, because it has a vanishing imaginary stability boundary. Now consider the 'smoothed Runge method'

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t\, \mathbf{f}(t_n + \Delta t/2,\ S\mathbf{y}_n + \Delta t\, \mathbf{f}(t_n, \mathbf{y}_n)/2). \tag{4'}$$

If the smoothing matrix S is defined by (1), then this scheme is just the Richtmyer scheme which can be proved to be (conditionally) stable for a class of hyperbolic model problems (cf. Wilson [12]).

Examples of the stabilization of Lax-Wendroff and Runge-Kutta type methods by smoothing of intermediate solution values may be found in a paper of de Goede [3].

1.2. Residue-smoothing

Instead of smoothing the numerical solution, one may consider smoothing of the residue of the difference scheme, that is, the high frequency components occurring in the residue vector (which is left upon substitution of the current numerical approximation into the difference scheme) are damped rather than the high frequency components of the solution. This approach is often followed in methods for solving elliptic boundary-value problems. A familiar example is the relaxation process employed in multigrid methods which is essentially a smoothing process in order to get rid of the high frequencies of the residue. Popular relaxation (or smoothing) methods are based on Gauss-Seidel iteration and incomplete LU decompositions.

The idea of residue-smoothing has also been used in (explicit) time-stepping methods for finding the stationary solution of time-dependent partial differential equations. In fact, such methods may be considered as iterative methods for solving boundary-value problems, so that it is not surprising that residue-smoothing is effective here too. More or less recent contributions are those of Lerat [7], Jameson [6], and Turkel [11]. In these papers, some form of *implicit* smoothing is applied, i.e., if $\mathbf{f}$ is the vector to be smoothed, then the vector $\mathbf{g}$ is obtained by solving the equation $R\mathbf{g} = \mathbf{f}$, where R is a sufficiently simple matrix so that it can 'conveniently' be solved.

An *explicit* smoothing process for smoothing the residue in Jacobi-type iteration, and thereby accelerating the convergence substantially, is considered in van der Houwen, Boon and Wubs [4]. A similar smoothing technique can be applied to the residues occurring in predictor-corrector methods for integrating semi-discrete parabolic problems. The stability of smoothed predictor-corrector methods is

analysed in van der Houwen and Sommeijer [5]. In Section 3, an example of such a method is reproduced.

1.3. Smoothing of the right-hand side function

An interesting approach for stabilizing explicit time integrators has been proposed by Wubs [13]. In this paper, the semi-discrete system of ordinary differential equations (3) is replaced by its 'smoothed' version

$$(3') \qquad dy(t) / dt = S\mathbf{f}(t, \mathbf{y}(t)),$$

where the matrix S is such that the spectral radius of the matrix $SJ := S\partial\mathbf{f}/\partial\mathbf{y}$ is much smaller than the spectral radius of the Jacobian matrix $J := \partial\mathbf{f}/\partial\mathbf{y}$ of the original system (3). Since the severe stability condition of explicit time integrators when applied to parabolic or hyperbolic problems is due to the large magnitude of the spectral radius associated with the right-hand side function, the stability condition corresponding to time integrators applied to (3') is considerably less restrictive than the condition corresponding to the same time integrators applied to (3). In the paper of Wubs, smoothing matrices are developed which allow an extremely simple implementation, and which are particularly efficient on vector computers.

A drawback of replacing (3) by (3') is a possible error introduced by the matrix S. Assuming that along the solution y(t), the right-hand side function is smooth with respect to the space variable x (does not contain high frequencies), the matrix S should be such that the magnitude of $\| S - I \|$, where the norm $\| \|$ is taken with respect to the subspace spanned by the lower frequencies, is sufficiently small. The development of explicit smoothing matrices tuned to the two-dimensional shallow water equations and a full analysis of stability and accuracy aspects is given in Wubs [14].

Certain methods belonging to the class of 'generalized' Runge-Kutta methods may also be interpreted as 'smoothed right-hand side' integration methods. Generalized Runge-Kutta methods are Runge-Kutta methods of which the parameters are replaced by matrices. For example, consider the first-order method

$$(5) \qquad y_{n+1} = y_n + \Delta t\, Q(\Delta t J)\, \mathbf{f}(t_n, y_n),$$

where Q(z) is a rational function satisfying the condition

$$Q(z) = 1 + c\, z + O(z^2) \quad \text{as} \quad z \to 0$$

with c some bounded constant, and where J is evaluated at some step point depending on the rate of variation with t. In passing we observe that we have second-order accuracy for c = 1/2 provided **f** does not depend on t and J is evaluated at (t_n, y_n).

The one-step method (5) may be interpreted as an explicit Euler method applied to the smoothed equation (3') with smoothing matrix $S = Q(\Delta t J)$. The function Q(z) is related to the stability function R(z) of the method by the relation

$$R(z) = 1 + z\, Q(z).$$

Consider the case where the system (3) originates from a parabolic problem, so that J has (more or less) negative eigenvalues, and suppose that an explicit integration

method is desired. It is well known that suitable stability functions for explicit one-step methods are provided by the shifted Chebyshev polynomials

$$R(z) = T_k(1+ z/k^2),\ T_k(x) := \cos(k \arccos(x)).$$

One-step methods of Taylor and Runge-Kutta type employing these polynomials have frequently been considered in the literature (cf., e.g.,Yuan'Chzao-Din [15], Franklin [1], Metzger [9], Lomax [8], Gentzsch & Schlueter [2]). If we introduce these polynomials in (5),we obtain

$$Q(z) = [R(z) - 1] / z = [T_k(1+ z/k^2) - 1] / z, \tag{6}$$

so that the matrix S becomes

$$S = Q(\Delta tJ) = [T_k(1+ \Delta tJ /k^2) - I]\ \lceil \Delta tJ \rceil^{-1}.$$

If Δt is chosen as large as allowed by the real stability boundary of the stability polynomial R(z), i.e. $\Delta t = 2k^2/\rho(J)$, then

$$S = [T_k(1+ 2J / \rho(J)) - I]\ \ [2k^2 J / \rho(J)]^{-1}. \tag{7}$$

It is easily verified that the spectral radius of the matrix SJ is a factor k^2 smaller than the spectral radius of the original Jacobian matrix J. The 'smoothed' Euler method defined by (5) and (6) is of practical interest if the computation of and the premultiplication by the matrix $S = Q(\Delta tJ)$ is inexpensive.

2. Smoothing matrices

We shall construct smoothing matrices S for vectors **f** which are discretizations of functions f(**x**), and we briefly discuss their implementation. Furthermore, we consider the effect of smoothing matrices when used as preconditioners.

2.1. Definition and properties

Let D be a difference matrix with a complete eigensystem. If this system is denoted by $\{e_j\}$,then the vector **f** to be smoothed can be represented in the form

$$\mathbf{f} = \sum_j c_j e_j. \tag{8}$$

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, and which satisfies the condition

$$P(z) = 1 + O(z^r) \text{ as } z \to 0 \text{ with } r > 0. \tag{9}$$

If P(z) satisfies (9), then the smoothing matrix P(D) will be called *consistent of order* r. Since,

(10) $$P(D)\mathbf{f} = \sum_j P(\mu_j)c_j\mathbf{e}_j,\ \mu_j := \text{eigenvalue of } D,$$

we see that the higher frequencies are increasingly stronger damped; the lower frequencies are less damped as r is larger. Ideally, the construction of the smoothing matrix P(D) should be simple and inexpensive. This can be achieved if the matrix D possesses a simple structure. For example, if **f** originates from a one-dimensional grid function defined on a uniform grid of width h, that is, there exist functions interpolating the successive components of **f** which are sufficiently smooth as h -> 0, then suitable matrices are the difference matrices defined by

(11) $$D = (1/4)\begin{bmatrix} 0 & & & & 0 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 0 & & & & 0 \end{bmatrix},\quad D = (1/2)\begin{bmatrix} 0 & & & & 0 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ 0 & & & & 0 \end{bmatrix},$$

respectively with eigenvalues in the interval [-1,0] and [-i,i].

The most simple polynomial satisfying the condition of first-order consistency is given by P(z) = 1+ az. The first difference matrix in (11) then leads to the smoothing matrix

$$P(D) = (1/4)\begin{bmatrix} 4 & & & & 0 \\ a & 4-2a & a & & \\ & \ddots & \ddots & \ddots & \\ & & a & 4-2a & a \\ 0 & & & & 4 \end{bmatrix}.$$

For a=2, this matrix is easily recognized as the smoothing matrix S defined by (1). Generally, P(D) has the property that the first and last component of the vector to which it is applied remain unchanged, provided that P(0) = 1.The generalization to more dimensions of the matrices D defined by (11), and the corresponding smoothing matrices P(D), is straightforward.

The following theorem provides polynomials derived from Chebyshev polynomials of the *first* kind, which can be used in combination with arbitrary difference matrices D with *real* eigenvalues (cf. [4]).

Theorem 1. *Let* P(z) *be defined by*

(12) $$P(z) := [T_k(1+2z) - 1] / [2zk^2],$$

and let D *has its eigenvalues* μ_j *in the interval* [-1,0]. *Then the following holds*:

(a) *The smoothing matrix* P(D) *is first-order consistent and its eigenvalues satisfy the inequality*

$$0 \le P(\mu_j) \le \min\{1, -1/k^2\mu_j\}.$$

(**b**) *Let the factor matrices* F_j *be generated by*

$$F_1 := I + D,\ F_{j+1} := (I - 2F_j)^2,\ j>0, \tag{13a}$$

and let $k = 2^q$, *then* $P(D)\mathbf{f}$ *can be obtained by the factorization formula*

$$P(D)\mathbf{f} = F_q F_{q-1} \ldots F_1 \mathbf{f}. \tag{13b}$$ □

Notice that for $D = J / \rho(J)$, the matrix $P(D)$ defined by (12) becomes identical with the smoothing matrix defined by (7).

The analogue of Theorem 1 for the case where D has *imaginary* eigenvalues employs Chebyshev polynomials of the *second* kind, and is given by

Theorem 2. *Let* $P(z)$ *be defined by*

$$\begin{aligned} P(z) &:= U_{2k}([1 + z^2]^{1/2}) / [2k + 1], \\ U_{2k}(x) &:= \sin((2k+1) \arccos(x))/\sin(\arccos(x)), \end{aligned} \tag{14}$$

and let D *has its eigenvalues* μ_j *in the interval* $[-i,i]$. *Then the following holds*:

(**a**) *The smoothing matrix* $P(D)$ *is first-order consistent and its eigenvalues satisfy the inequality*

$$|P(\mu_j)| \le \min \{1,\ 1 / (2k+1)\ |\mu_j|\}.$$

(**b**) *Let the factor matrices* F_j *be generated by*

$$F_1 := 2[I + 2D^2],\ F_{j+1} := (F_j)^2 - 2I,\ j>0, \tag{15a}$$

and let $2k = 2^q$, *then* $P(D)\mathbf{f}$ *is recursively defined by*

$$\begin{aligned} \mathbf{f}_1 &= (I + F_1)\mathbf{f};\quad \mathbf{f}_{j+1} = \mathbf{f} + F_j\mathbf{f}_j,\ j = 1, \ldots, q-1; \\ P(D)\mathbf{f} &= (2k+1)^{-1}\mathbf{f}_q. \end{aligned} \tag{15b}$$ □

Part (**a**) of these theorems characterizes the damping power of P(D) on the high frequencies. Part (**b**) presents P(D) in factorized form which is of importance in the implementation of smoothing matrices.

2.2. Implementation of smoothing matrices

We mention three possibilities for implementing the smoothing matrix P(D). For details we refer to [4].

(i) The most straightforward (but rather storage consuming) way consists of computing the elements of the matrix P(D) in advance. This is feasible for simply structured D in one spatial dimension, but in more dimensions it is not recommendable. However, what one might do in more dimensional problems, is the application of the one-dimensional smoothing matrices successively in the spatial directions.

(ii) An alternative is the use of the recursive relations satisfied by Chebyshev polynomials. If k is the degree of the polynomial P(z), then the generaration of a smoothed vector $\mathbf{g} = P(D)\mathbf{f}$ 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.

(iii) Finally, we mention the most interesting implementation method which is based on the factorization property presented in part (**b**) of the theorems above. If the factor matrices defined in (13a) and (15a) are computed in advance, then the generation of the smoothed vector $P(D)\mathbf{f}$ requires, in both of the cases (13b) and (15b) only $q = {}^2\log(k)$ matrix-vector multiplications. The reduced number of matrix-vector multiplications is only an advantage if the factor matrices have relatively few nonzero elements in each row. For example, this is true if D is defined by the matrices given in (11). For the first matrix, we find the factor matrices

$$F_1 = (1/4)\begin{bmatrix} 4 & & & & 0 \\ 1 & 2 & 1 & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & 1 & 2 & 1 \\ 0 & & & & & 4 \end{bmatrix}, \quad F_2 = (1/4)\begin{bmatrix} 4 & & & & & 0 \\ 2 & 1 & 0 & 1 & & \\ 1 & 0 & 2 & 0 & 1 & \\ & \ddots & & & \ddots & \\ & & 1 & 0 & 2 & 0 & 1 \\ & & & 1 & 0 & 1 & 2 \\ 0 & & & & & & 4 \end{bmatrix}, \text{ etc.}$$

In the case where D is defined by the second matrix given in (11) we obtain the factor matrices

$$F_1 = \begin{bmatrix} 2 & & & & 0 \\ 0 & 1 & 0 & 1 & \\ 1 & 0 & 0 & 0 & 1 \\ & \ddots & & & \\ & & 1 & 0 & 0 & 0 & 1 \\ & & & 1 & 0 & 1 & 0 \\ 0 & & & & & & 2 \end{bmatrix}, \quad F_2 = \begin{bmatrix} 4 & & & & & & & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 & & \\ 2 & 0 & 1 & 0 & 0 & 0 & 1 & \\ 0 & 1 & 0 & 2 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \\ & & & & \ddots & & & & & \end{bmatrix}, \text{ etc.}$$

Evidently, matrix-vector multiplications with these essentially three-diagonal factor matrices is extremely cheap, particularly on vector computers where so-called linked triads can be used.

The precomputation of the factor matrices seems to be feasible only in one-dimensional cases. In more dimensional problems, one might follow the approach of applying one-dimensional matrices in the successive spatial directions mentioned above (see (i)).

2.3. Preconditioning by smoothing matrices

Preconditioning by smoothing matrices is here defined as the premultiplication of a given vector function $\mathbf{f}(\mathbf{y})$ of a vector variable $\mathbf{y}$ by a smoothing matrix S (*preconditioning matrix*) such that the condition of the Jacobian of the new function (that is, the matrix SJ) is better than the condition of the Jacobian of the original function (the matrix J). Preconditioning by smoothing matrices consists of constructing smoothing matrices S such that the spectrum condition number of the matrix SJ is substantially reduced. An attractive property of smoothing preconditioners is that they are, to a large extent, independent of the function to which they are applied. This is in contrast to most other preconditioners, such as preconditioning by SSOR or by incomplete factorization.

In the case of *explicit* smoothing procedures, the matrix S may be chosen of the form $S = P(D)$, where D is a difference matrix and P(z) is a polynomial of one of the forms derived above. Ideally, the matrix D is chosen equal to the Jacobian of $\mathbf{f}$ and normalized by its spectral radius, i.e.,

$$D = [\rho(J)]^{-1} J. \tag{16}$$

The eigenvalues of SJ are then generated by the polynomial

$$\rho(J)\, z\, P(z),$$

where z runs through the spectrum of D. In actual computation, however, it is generally not attractive to choose D as defined by (16), and one employs some cheap approximation to the normalized Jacobian matrix of $\mathbf{f}$. In choosing a cheap difference matrix D it seems recommendable to take into account the type of eigenvalue spectrum of J. For instance, if $\mathbf{f}(\mathbf{y})$ represents the discretization of some second-order (nonlinear) differential operator applied to a one-dimensional grid function $\mathbf{y}$, then J has often negative eigenvalues, and one may try the difference matrix given by the first matrix in (11). Likewise, in the case of first-order differential operators, J has often purely imaginary eigenvalues, and one may try the second difference matrix in (11).

In order to see the effect on the spectrum condition number of J when premultiplied by a smoothing matrix we shall assume that we are in the ideal situation where (16) is satisfied. From Theorem 1 and 2 we obtain

Theorem 3. (**a**) *Let* D *be defined by* (16), *let the eigenvalues of* J *be real nonpositive, and let* $S := P(D)$, *where* P(z) *is defined by* (12). *Then the eigenvalues of* SJ *are again real nonpositive and, of all first-order consistent polynomials which are nonpositive in* [-1,0], *the polynomial* (12) *minimizes the spectral radius of* P(D)D.

This spectral radius is given by

$$\rho(SJ) = \rho(J) / k^2.$$

(b) *Let* D *be defined by* (16), *let the eigenvalues of* J *be purely imaginary (zero not excluded), and let* S := P(D), *where* P(z) *is defined by* (14). *Then the eigenvalues of* SJ *are again purely imaginary and of all first-order consistent polynomials, the polynomial* (14) *minimizes the spectral radius of* P(D)D. *This spectral radius is given by*

$$\rho(S\partial f/\partial y) = \rho(\partial f/\partial y) / (2k + 1). \text{ []}$$

3. Smoothed predictor-corrector methods

3.1. The computational scheme

One of the many possible applications of the smoothing matrices described above is the construction of highly stable predictor-corrector methods for semi-discrete parabolic or hyperbolic initial-value problems. The following $P(EC)^m$ type method is based on the linear extrapolation predictor and the second-order backward differentiation corrector, and is designed for the integration of parabolic equations (cf. [5]):

For all m $\quad y^{(0)} = 2y_n - y_{n-1}$;

if m = 1 then $\quad y_{n+1} = y^{(0)} - \mathbf{S}\mathbf{r}(y^{(0)})$

(17) if m > 1 then $\quad y^{(1)} = y^{(0)} - c\mathbf{S}\mathbf{r}(y^{(0)})$,

$$y^{(j)} = 2y^{(j-1)} - y^{(j-2)} - 2c\mathbf{S}\mathbf{r}(y^{(j-1)}),\ j = 2, \ldots, m-1,$$

$$y_{n+1} = [y^{(0)} - 2\,y^{(m-2)} + 4y^{(m-1)} - 4c\mathbf{S}\mathbf{r}(y^{(m-1)})]\,/3.$$

Here, $\mathbf{r}(y)$ is the residue function associated with the corrector:

(18) $$\mathbf{r}(y) := [3y - 2\Delta t\,\mathbf{f}(t_{n+1}, y) - 4y_n + y_{n-1}]\,/3,$$

and c is a constant given by

(19) $$c := 1 - \cos(\arccos(-1/2)/m) \approx 1 - \cos(2.094 / m).$$

We observe that this method reduces to the conventional predictor-corrector method in $P(EC)^m$.

The following theorem characterizes the method in the model case where (16) is satisfied.

Theorem 4. *Let* D *be defined by* (16) *and let* S := P(D), *where* P(z) *is defined by* (12). *Then the method defined by* (17) - (19) *is second-order accurate and the real stability boundary satisfies the inequality*

$$\beta_{real} > (3/2 + \text{int}\,[3(2 - c) / 2c])k^2 - 3 / (1 - \cos(\pi / k)),$$

where int(.) *denotes the integer-part function.* []

For large values of m and k the real stability boundary behaves as

$$\beta_{real} \approx 1.37\, m^2 k^2. \tag{20}$$

As m+k increases, this approximation decreases rapidly to the true value of the stability boundary. For m+k=4, the error is already less than10%.

We recall that, in actual computation, we do not define the matrix D by (16), but, in the case of Jacobian matrices with real eigenvalues, by the first matrix defined in (11). Extensive numerical experiments have shown that the stability boundary given in the above theorem is still reliable (cf. [5]). Furthermore, we should always choose $k = 2^q$ in order to exploit the factorized implementation of the corresponding smoothing matrices (only q three-diagonal matrix-vector multiplications for each spatial dimension). Thus, the computational costs required to perform a step

$$\Delta t = \beta_{real} / \rho(J) \approx 1.37\, m^2 4^q / \rho(J) \tag{21}$$

are m smoothed right-hand side evaluations. This expression shows that we can take any step we want by choosing q and m sufficiently large. Since increasing q is relatively inexpensive and increasing m is usually very expensive, it is tempting to set m=1 and to increase only q. However, in our experiments we observed a systematic drop of accuracy for large values of q (cf. [5]).

3.2. Computational costs

We conclude this paper with a derivation of the computational costs for a d-dimensional parabolic problem defined on the unit box in (t,x)-space and discretized on a grid with mesh size h both in the time direction and in all spatial directions. Let the spectral radius for this problem be dr/h^2, r being a constant, let $\Delta t = h$ and set m=m*. Then (21) yields

$$q \geq q^*,\ q^* := {}^4\log(1/h) - 2\,{}^4\log(m^*) + s,\ s := {}^4\log(dr/1.37). \tag{22}$$

The total computational costs are given by

$$C(h) = m^*[\varphi + dq\varepsilon]\, N,\ N := h^{-(d+1)}, \tag{23}$$

where φ and ε respectively denote the costs per grid point involved for evaluating f and for performing a factor matrix multiplication, and where N denotes the total

number of grid points in the (d+1)-dimensional box. Substitution of $q = q^*$ into (23) yields

$$C(h) \approx m^*N\,[\varphi + d\varepsilon\,(^4\log(h^{-1}) - 2^4\log(m^*) + s)] = N\,O(\log(h^{-1})).$$

References

[1] Franklin, J.N.: Numerical stability in digital and analogue computation for diffusion problems, J. Math. Phys. 37, 305-15 (1959).

[2] Gentzsch, W. & Schlueter, A.: Ueber ein Einschrittverfahren mit zyklischer Schrittweitenanderung zur løsung parabolischer Differentialgleichungen, ZAMM 58, T415-T416 (1978).

[3] Goede, E. de: Stabilization of the Lax-Wendroff methods and a generalized one-step Runge-Kutta method for hyperbolic initial-value problems, Report NM-R8613, Centre for Mathematics and Computer Science, Amsterdam (1986), to appear in Appl. Numer. Math..

[4] Houwen, P.J. van der, Boon, C. & Wubs, F.W.: Analysis of smoothing matrices for the preconditioning of elliptic difference equations, Report NM-R8705, Centre for Mathematics and Computer Science, Amsterdam (1987), to appear in ZAMM.

[5] Houwen, P.J. van der & Sommeijer, B.P.: Improving the stability of predictor-corrector methods by residue smoothing, Report NM-R8707, Centre for Mathematics and Computer Science, Amsterdam (1987).

[6] Jameson, A.: The evolution of computational methods in aerodynamics, J. Appl. Mech. 50, 1052-76 (1983).

[7] Lerat, A.: Une class de schemas aux difference implicites pour les systemes hyperboliques de lois de conservation, C.R. Acad. Sc. Paris, t. 288 (Serie A), 1033-36 (1979).

[8] Lomax, H.: On the construction of highly stable, explicit numerical methods for integrating coupled ODEs with parasitic eigenvalues, NASA Technical Note NASAIN D/4547 (1968).

[9] Metzger, C.L.: Methodes de Runge-Kutta de rang superieure a l'ordre, These (Troisieme cycle), Universite de Grenoble (1967).

[10] Shuman, F.G.: Numerical methods in weather prediction II. Smoothing and filtering, Monthly Weather Review 85, 357-371 (1957).

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

[12] Wilson, J.C.: Stability of Richtmyertype difference schemes in any finite number of space variables and their comparison with multistep Strang schemes, J. Inst. Math. Applics. 10, 238-257 (1972).

[13] Wubs, F.W.: Stabilization of explicit methods for hyperbolic partial differential equations, Int. J. Numer. Methods Fluids 6, 641-57 (1986).

[14] Wubs, F.W.: Numerical solution of the shallow water equations, Thesis, University of Amsterdam (1987).

[15] Yuan' Chzao-Din: Some difference schemes for the solution of the first boundary value problem for linear differential equations with partial derivatives, Thesis, Moskow State University (1958).