IMA Journal of Numerical Analysis (1986) **6**, 1–23

On the Stability of Predictor–Corrector Methods for Parabolic Equations with Delay

P. J. VAN DER HOUWEN AND B. P. SOMMEIJER
Centre for Mathematics and Computer Science, Kruislaan 413, Amsterdam

AND

CHRISTOPHER T. H. BAKER
University of Manchester

[Received 24 August 1984 and in revised form 28 February 1985]

Diffusion problems where the current state depends upon an earlier one give rise to parabolic equations with delay. The efficient numerical solution of classical parabolic equations can be accomplished via methods for stiff differential equations; one such class is predictor–corrector-type methods with extended real stability intervals and with reduced storage requirements. Analogous methods for equations with delay are proposed and analysed here. Our analysis will be based on the test equation $\dot{y}(t) = q_1 y(t) + q_2 y(t-\omega)$, where, in view of the class of parabolic delay equations we want to consider, our main interest will be in the case $|q_1| \gg |q_2|$. Implementational details of the methods developed are given and numerical results are presented.

1. Introduction

THERE IS an extensive literature on the theory and numerical solution of parabolic equations. The inclusion of a delay in the classical problems of mathematical physics leads to partial differential equations with delay only in time, t. As illustration, consider the *generalized diffusion equation*

$$\frac{\partial}{\partial t}\boldsymbol{u}(t,x) = \alpha^2 \frac{\partial^2}{\partial x^2}\boldsymbol{u}(t,x) + \boldsymbol{f}\big(\boldsymbol{u}(t-\omega,x)\big) \quad (0 \leqslant x \leqslant 1, \quad t \geqslant \omega) \tag{1.1}$$

with homogeneous Dirichlet conditions on $x = 0$ and $x = 1$ and the prescribed initial function

$$\boldsymbol{u}(t,x) = \boldsymbol{\phi}(t,x) \quad (0 \leqslant t \leqslant \omega, \quad 0 \leqslant x \leqslant 1). \tag{1.2}$$

The existence and uniqueness theory for problems of this type has been discussed by Travis & Webb (1974), for example, in the case that $\boldsymbol{f}$ is a linear or nonlinear scalar-valued function. Cases where the term $\boldsymbol{f}$ is replaced by more general expressions involving the state $\boldsymbol{u}(t-\omega, x)$ also arise (for example see El'sgol'ts & Norkin, 1973, pp. 269–272):

$$\frac{\partial}{\partial t}\boldsymbol{u}(t,x) = \alpha^2 \frac{\partial^2}{\partial x^2}\boldsymbol{u}(t,x) + \beta^2 \frac{\partial^2}{\partial x^2}\boldsymbol{u}(t-\omega,x); \tag{1.3}$$

the theory of a class of examples of general type is discussed by Artola (1967).

Wang (1963) provides an example of a realistic system (an automatically controlled furnace): the system is modelled by an equation which falls into the class of problems of the form

$$\frac{\partial}{\partial t}\boldsymbol{u}(t,x)=L\boldsymbol{u}(t,x)+\boldsymbol{f}\left(t,x,\boldsymbol{u}(t,x),\boldsymbol{u}(t-\omega^{(1)},x),\ldots,\frac{\partial \boldsymbol{u}}{\partial x}(t,x),\frac{\partial \boldsymbol{u}}{\partial x}(t-\omega^{(2)},x),\ldots\right) \tag{1.4}$$

involving multiple delays $\omega^{(1)}$, $\omega^{(2)}$, etc. where L is a linear operator which is uniformly elliptic in x.

Time delays can enter into diffusion systems in various ways. Wang (1975) considers realistic systems in which the delay term is absent from the differential equation but enters into the boundary conditions valid for $x=0$, 1 and $t\geqslant 0$.

In the case $\boldsymbol{f}\equiv\boldsymbol{0}$ in (1.1), numerical methods for the approximation of $\boldsymbol{u}(t,x)$ can be derived, as is well known (see also e.g. Lambert, 1973, p. 249), by semidiscretization in the x-variable and the numerical solution of the resulting ordinary differential equations with respect to time. In the generalized equations considered above, the process of semidiscretization produces (in place of a system of ordinary differential equations) a system of retarded differential equations. Thus, the simplest discretization scheme yields for (1.1) the equations

$$\dot{\mathbf{y}}_i(t)=\frac{\alpha^2}{h^2}[\mathbf{y}_{i+1}(t)-2\mathbf{y}_i(t)+\mathbf{y}_{i-1}(t)]+\boldsymbol{f}(\mathbf{y}_i(t-\omega),ih) \tag{1.5}$$

and that for (1.3) yields

$$\dot{\mathbf{y}}_i(t)=\frac{\alpha^2}{h^2}[\mathbf{y}_{i+1}(t)-2\mathbf{y}_i(t)+\mathbf{y}_{i-1}(t)]+\frac{\beta^2}{h^2}[\mathbf{y}_{i+1}(t-\omega)-2\mathbf{y}_i(t-\omega)+\mathbf{y}_{i-1}(t-\omega)], \tag{1.6}$$

where $\mathbf{y}_i(t)\approx\boldsymbol{u}(t,ih)$, $h=1/(N+1)$, $i=1,\ldots,N$.

In the case of a general linear problem, involving one delay, the semidiscretization process yields a system of equations of the form

$$\dot{\mathbf{y}}(t)=Q_1\mathbf{y}(t)+Q_2\mathbf{y}(t-\omega) \tag{1.7}$$

where $\mathbf{y}(t)=[\mathbf{y}_1^{\mathrm{T}}(t),\ldots,\mathbf{y}_N^{\mathrm{T}}(t)]^{\mathrm{T}}$. In the cases where the matrices Q_i are simultaneously diagonalizable (which occurs for (1.5) where $\boldsymbol{f}(\mathbf{y}_i,ih)=\mathbf{y}_i$, and for (1.6)) the study of scalar test equations of the form

$$\dot{y}(t)=q_1y(t)+q_2y(t-\omega) \tag{1.8}$$

provides insight concerning the behaviour as $t\to\infty$ of solutions of (1.7). If in (1.5) h is small and in (1.6) $\alpha^2\gg\beta^2$, then the corresponding equation (1.8) has the property that $|q_1|\gg|q_2|$. In our discussion of stability such test equations will occupy our attention in particular.

Retarded differential equations are derived here through semidiscretization but also arise directly in their own right in various applications; see Chosky (1966), Weiss (1959). It is well known that the efficient numerical solution of the ordinary differential equations obtained on semidiscretization of (1.1) with $\boldsymbol{f}\equiv\boldsymbol{0}$ requires numerical methods with large regions of stability. Generalized predictor–corrector methods with extended region of stability have been derived and studied by van

der Houwen & Sommeijer (1983b) with this application in mind. van der Houwen & Sommeijer (1983a) adapted their numerical methods to ordinary differential equations with delay, of the general form $\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t), \mathbf{y}(t-\omega))$ with $\omega > 0$. Since such methods are well-suited to the numerical solution of the retarded differential equations obtained on semidiscretization of generalized diffusion equations, we develop the results of van der Houwen & Sommeijer (1983b) with this application in mind.

2. Predictor–corrector methods

In this section we will discuss the construction of numerical methods by reference to a general nonlinear system of delay equations involving one delay, that is,

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t), \mathbf{y}(t-\omega)), \qquad \omega \equiv \omega(t, \mathbf{y}(t)) \geq 0, \qquad t \geq t_0, \tag{2.1}$$

with $\mathbf{y}(t)$ prescribed at (and, if necessary, on an interval to the left of) the point t_0. In the present section we assume only such conditions as ensure smoothness of $\mathbf{f}$ and the existence of a unique (smooth) solution $\mathbf{y}(t)$. (Later, we assume that the Jacobian matrices of derivatives of $\mathbf{f}(t, \mathbf{u}, \mathbf{v})$ with respect to $\mathbf{u}$ and to $\mathbf{v}$ have the same eigensystem and real eigenvalues.)

The methods we describe are predictor-corrector methods for use with formulae discussed by Cryer (1974); they reduce, in the case that the delay term is absent, to methods considered in van der Houwen & Sommeijer (1983b) for the initial-value problem

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t)), \qquad t \geq t_0 \tag{2.1'}$$

with $\mathbf{y}(t_0)$ prescribed.

We denote by $\{\rho, \sigma\}$ the implicit linear multistep formula (cf. Lambert, 1973, pp. 11–43) with first and second characteristic polynomials

$$\rho(\zeta) = \sum_{i=0}^{k} a_i \zeta^{k-i} \quad \text{and} \quad \sigma(\zeta) = \sum_{i=0}^{k} b_i \zeta^{k-i}.$$

We shall call this formula *the corrector formula*. We assume that the corrector is zero-stable, consistent and of order p (Lambert, 1973, p. 23). We shall denote by $\{\tilde{\rho}, \tilde{\sigma}\}$ a corresponding explicit formula (*the predictor formula*, with $b_0 = 0$), and its order by $\tilde{p}$.

It is necessary to adapt the formulae for (2.1′) to permit the treatment of (2.1). Our objective, given a constant integration step $\Delta t > 0$, is to approximate the solution $\mathbf{y}(t_n)$ of (2.1) at $t_n = t_0 + n\Delta t$ by $\mathbf{y}_n$ $(n = 1, 2, 3, \ldots)$; for this purpose we shall approximate $\mathbf{y}(t_n - \omega_n)$, with $\omega_n = \omega(t_n, \mathbf{y}_n)$, using polynomial interpolation on the values $\mathbf{y}_j, \mathbf{y}_{j-1}, \ldots, \mathbf{y}_{j-l}$ where $t_{j-1} < t_n - \omega_n \leq t_j$ for $j > 0$. Usually, Hermite interpolation is employed; however, in view of the application to parabolic equations we have in mind, we shall use one of those backward differentiation formulae which are highly stable as corrector; consequently, no $\mathbf{f}$-values are stored, preventing us from using Hermite interpolation. The interpolation formula

assumes the form

$$\hat{\mathbf{y}}(t_n-\omega_n)=\mathsf{E}^{-l}\tau(\mathsf{E},\theta_n)\mathbf{y}_j \tag{2.2}$$

where $t_n-\omega_n=t_j-\theta_n\Delta t$ with $0\leqslant\theta_n<1$ and E is the forward shift operator $\mathsf{E}\boldsymbol{\phi}_n=\boldsymbol{\phi}_{n+1}$. Here, τ is a polynomial in E whose coefficients depend upon θ_n and (2.2) is merely a symbolic form of Newton's backward formula. Concerning τ we assume $\tau(\xi,0)=\zeta^l$ and $\tau(1,\theta_n)\equiv 1$; the order of accuracy of (2.2) is $l+1$. We shall assume that the order of the interpolation formula (2.2) is at least that of the method $\{\rho,\sigma\}$ for (2.1′), i.e. $l\geqslant p$.

The formulae which form the basis for the numerical method for (2.1) now comprise (2.2) and

$$\rho(\mathsf{E})\mathbf{y}_n-\Delta t\sigma(\mathsf{E})\mathbf{f}_n=\mathbf{0}\quad(n\geqslant 0);\qquad \mathbf{f}_n=\begin{cases}\mathbf{f}(t_n,\mathbf{y}_n,\hat{\mathbf{y}}(t_n-\omega_n)) & (t_n-\omega_n>t_0),\\ \mathbf{f}(t_n,\mathbf{y}_n,\mathbf{y}(t_n-\omega_n)) & (t_n-\omega_n\leqslant t_0);\end{cases}$$
$$\omega_n=\omega_n(t_n,\mathbf{y}_n). \tag{2.3}$$

We refer to (2.3) as *the delay–corrector formula.*

Since $b_0\neq 0$ the formulae (2.3) are certainly implicit. At each integration step it is necessary to solve

$$a_0\mathbf{y}_n-b_0\Delta t\mathbf{f}(t_n,\mathbf{y}_n,\hat{\mathbf{y}}_n(t_n-\omega_n))=\mathbf{w}_n,\qquad a_0=1, \tag{2.4}$$

coupled with (2.2), where $\mathbf{w}_n$ is computable in terms of values of $\mathbf{y}_k$ already computed. From now on we assume $a_0=1$ and $b_0>0$. Observe that $\hat{\mathbf{y}}(t_n-\omega_n)$ will, when $\omega_n<\Delta t$, depend on the as yet unknown approximation $\mathbf{y}_n$ to $\mathbf{y}(t_n)$.

In order to solve (2.4) we use the following predictor–corrector scheme:

$$\left.\begin{aligned}&\mathbf{y}_n^{(0)}:=\text{initial approximation to the exact solution } \boldsymbol{\eta}_n \text{ of (2.4),}\\ &\qquad\text{to be provided by a predictor formula,}\\ &\mathbf{y}_n^{(j)}:=\mu_j\mathbf{y}_n^{(j-1)}+(1-\lambda_j-\mu_j)\mathbf{y}_n^{(j-2)}+\lambda_j b_0\Delta t\mathbf{f}_n^{(j-1)}+\lambda_j\mathbf{w}_n\quad(j=1,\ldots,m),\\ &\mathbf{y}_n:=\mathbf{y}_n^{(m)},\end{aligned}\right\} \tag{2.5}$$

where $\lambda_1+\mu_1=1$ with λ_j and μ_j $(j>1)$ to be determined later. Here

$$\mathbf{f}_n^{(j-1)}:=\mathbf{f}\big(t_n,\mathbf{y}_n^{(j-1)},\hat{\mathbf{y}}\big(t_n-\omega_n(t_n,\mathbf{y}_n^{(j-1)})\big)\big).$$

In passing we observe that a conventional predictor–corrector method for (2.1), in P(EC)^{m}E-mode, is obtained if we choose $\mu_j=0$ and $\lambda_j=1$ $(j=1,2,\ldots,m)$. Another special case of (2.5) was considered by Stetter (1968).

The general predictor-corrector method (2.5) will be called a *GPC method*; it falls into a still more general class of methods presented in van der Houwen & Sommeijer (1983b). For our purposes, (2.5) has sufficient degrees of freedom; our aim is the construction of GPC methods which permit the choice of large Δt when applied to (1.8) with $|q_1|\gg|q_2|$, bearing in mind the applicability of such methods to the solution via semidiscretization of a class of parabolic equations with delay. Therefore, the parameters λ_j and μ_j, and the number of iterations m, will be chosen in such a way that a stable method results. Thus, the GPC method is, in the first place, an iteration scheme for approximating a stable method, rather than an iteration scheme for approximating the solution of (2.4).

Remark. In practical computations, the choice of $\{\lambda_j, \mu_j\}$ in (2.5) will be determined by local conditions, but we shall ignore this feature until Section 4.

2.1 *The Local Error*

In studying the accuracy of the GPC method (2.5) it is convenient to introduce the *iteration polynomials* $P_j(z)$ generated by

$$\begin{aligned} P_0(z) &= 1, \qquad P_1(z) = 1-\lambda_1+\lambda_1 b_0 z, \\ P_j(z) &= (\mu_j+\lambda_j b_0 z)P_{j-1}(z)+(1-\lambda_j-\mu_j)P_{j-2}(z) \quad (j=2,3,\ldots,m). \end{aligned} \tag{2.6}$$

Notice that $P_j(1/b_0)=1$ for all j. The polynomials $P_j(z)$ are uniquely associated with the iteration scheme (2.5).

Furthermore, we need the Jacobian matrix of the right-hand-side function $\boldsymbol{f}_n$. Let $\boldsymbol{g}(t, \boldsymbol{u}, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_l)$ be the function such that

$$\boldsymbol{f}(t_n, \mathbf{y}_n, \hat{\mathbf{y}}(t_n-\omega_n)) = \boldsymbol{g}(t_n, \mathbf{y}_n, \mathbf{y}_{n-1}, \ldots).$$

Recalling that if $\omega_n \equiv \omega(t_n, \mathbf{y}_n) < \Delta t$ then $\hat{\mathbf{y}}(t_n-\omega_n)$ depends upon $\mathbf{y}_n$, we define the Jacobian matrix

$$Z_n := \Delta t \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}(t_n, \boldsymbol{\eta}_n, \mathbf{y}_{n-1}, \ldots), \tag{2.7}$$

where $\boldsymbol{\eta}_n$ is the exact solution (assumed unique) of the delay–corrector formula (2.3). The local error of the GPC method can be expressed in terms of the corresponding errors of the corrector and predictor formula using the iteration polynomial $P_m(z)$ and the matrix Z_n.

THEOREM 2.1 *If $\boldsymbol{f}(t, \boldsymbol{u}, \boldsymbol{v})$ and the solution $\mathbf{y}(t)$ are sufficiently smooth, then, provided $l \geq \max\{p, \tilde{p}\}$,*

$$\mathbf{y}_n - \mathbf{y}(t_n) = [I-P_m(Z_n)][\boldsymbol{\eta}_n - \mathbf{y}(t_n)] + P_m(Z_n)[\mathbf{y}_n^{(0)} - \mathbf{y}(t_n)] + O(\Delta t^{2p+3} + \Delta t^{2\tilde{p}+3}),$$

where $\tilde{p}$ and p are the orders of accuracy of the predictor formula for $\mathbf{y}_n^{(0)}$ and the corrector formula for $\boldsymbol{\eta}_n$, respectively, and where we assume $\mathbf{y}_j = \mathbf{y}(t_j)$ for $j<n$. □

The proof of this theorem is a slight elaboration of the proof of Theorem 3.1 and Corollary 3.1 given in van der Houwen & Sommeijer (1983b), and is therefore omitted.† From this theorem we immediately conclude that the order of the GPC method is given by $p^* = \min\{p+r, \tilde{p}+\tilde{r}\}$ where r is the multiplicity of the zero at $z=0$ of $1-P_m(z)$ and $\tilde{r}$ the multiplicity of the zero at $z=0$ of $P_m(z)$.

In actual applications the local error $\boldsymbol{\eta}_n - \mathbf{y}(t_n)$ of the corrector is usually small in comparison with the local error $\mathbf{y}_n^{(0)} - \mathbf{y}(t_n)$ of the predictor. Therefore, we will only consider polynomials with $r=0$ and choose $P_m(z)$ such that $P_m(Z_n)$ decreases the magnitude of the predictor error. If Δt is small, that is, $\|Z_n\|$ is small, this can be achieved by choosing $\tilde{r}$ as large as possible. For example, the conventional predictor–corrector method uses $P_m(z) = (b_0 z)^m$ so that $\tilde{r} = m$. However, we want to use relatively large integration steps and consequently (assuming

† The complete proof of Theorem 2.1 can be found in the institute report (same title, same authors) NM-R8410, Centre for Mathematics and Computer Science, Kruislaan 413 1098 SJ Amsterdam.

that the order terms in the statement of Theorem 2.1 remain negligible) we should choose $P_m(z)$ such that $|P_m(z)|$ is small in a sufficiently large neighbourhood of the origin on the negative z-axis. As we will see in the stability analysis of the GPC method, the stability condition also requires $|P_m(z)|$ to be small on a negative interval, $[-\beta, 0]$ say. Therefore, we postpone the choice of $P_m(z)$ to Section 3.2.

3. Stability theory

We mentioned previously, in Section 1, that the scalar test equation (1.8) provides insight concerning the behaviour as $t \to \infty$ of solutions of (1.7); we consider (1.7) as a linearization of (2.1) with Q_1 and Q_2 being defined as the Jacobian matrices $\partial \boldsymbol{f}/\partial \boldsymbol{u}$ and $\partial \boldsymbol{f}/\partial \boldsymbol{v}$ of $\boldsymbol{f}(t, \boldsymbol{u}, \boldsymbol{v})$, and assuming that Q_1 and Q_2 share the same eigensystem. The region in the real (q_1, q_2)-plane where the test equation (1.8) has solutions $y(t)$ such that $y(t) \to 0$ as $t \to \infty$, for a given delay ω, will be called the *stability region* corresponding to the delay ω. It can be shown (see e.g. Bellman & Cooke, 1963, p. 444) that in the real (q_1, q_2)-plane this open region is bounded by the curve

$$q_1 = q \cot \omega q, \qquad q_2 = -q/\sin \omega q, \tag{3.1}$$

parametrized by q with $0 \leq q \leq \infty$.

In Fig. 1 these curves are plotted in the $(q_1, q_2)\Delta t$-plane. To obtain the analytical stability region, which of course cannot have anything to do with Δt, the scaling factor Δt should be removed. However, this factor is included to facilitate comparisons with numerical stability regions, which are used to be plotted in the $(q_1 \Delta t, q_2 \Delta t)$-plane.

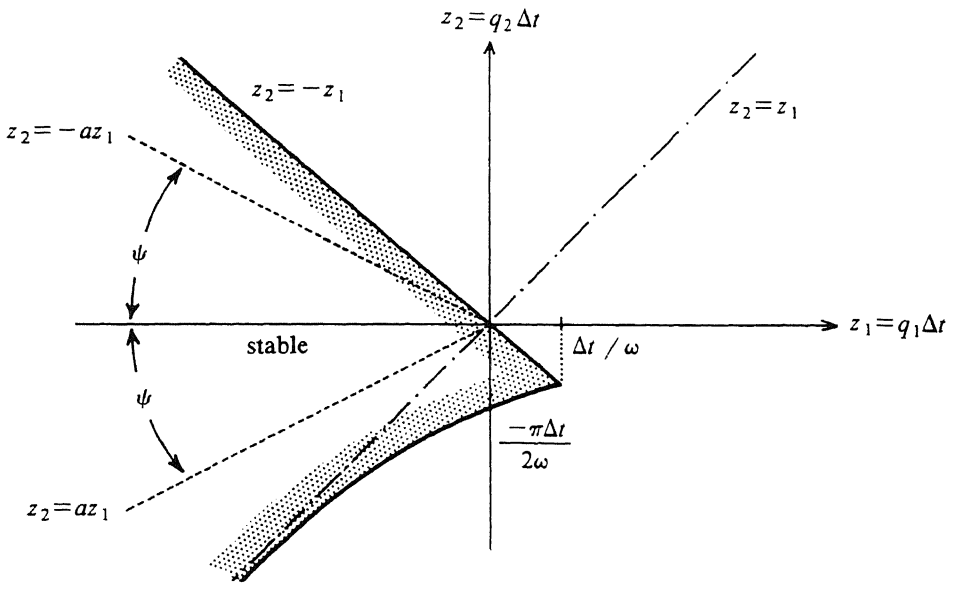


FIG. 1. Stability region of (1.8).

In analogy with the definition of the 'analytical' stability region (3.1) we define the numerical stability region as the set of points $(q_1, q_2)\Delta t = (z_1, z_2)$ for which the GPC method when applied to (1.8) yields solutions y_n such that $y_n \to 0$ as $n \to \infty$.

The GPC method is said to be *absolutely stable* for a given point (z_1, z_2) if this point lies in the stability region. (This terminology accords with the usage in e.g. Lambert (1973, p. 66), and is referred to as 'strict' in the writings of Baker (1977, p. 793) to distinguish it from the weaker definition sometimes encountered.) However, for brevity in what follows, we use 'stable at (z_1, z_2)'. If a method is stable at all points in the real infinite wedge $\{(z_1, z_2): z_1 < 0, |z_2/z_1| < a = \tan\psi\}$ then the method will be called $P_0(\psi)$-*stable* (see Figure 1). If the method is $P_0(\pi/4)$-stable we will briefly refer to P_0-stability (van der Houwen & Sommeijer, 1983a). In this connection it should be remarked that Barwell (1975) called a numerical method *P-stable* if the numerical stability region contains all complex points (z_1, z_2) with $\mathrm{Re}\, z_1 < -|z_2|$. The reader will note, on considering the case $z_2 = 0$, that $P_0(\psi)$-stable methods collapse in the case of equations with no delay to A_0-stable methods for pure differential equations. In consequence, $P_0(\psi)$-stable methods are necessarily implicit, whilst the GPC methods are explicit. It follows that the best we can expect is that the GPC methods have a region of stability which includes a truncated wedge $\{(z_1, z_2) \in \mathbb{R}^2 : -\beta < z_1 < 0, |z_2/z_1| < \tan\psi\}$. Such methods, with β large, we will term *almost-$P_0(\psi)$ stable.*

We will be particularly interested in almost-$P_0(\psi)$ stable methods with ψ small, because the semidiscrete parabolic delay equations discussed in Section 1 will lead to (z_1, z_2)-points located in a wedge $|z_2/z_1| < \tan\psi$ with small aperture 2ψ. The relevant range $[-\beta, 0)$ for z_1 is determined by the Jacobian matrices corresponding to the right-hand-side function and the discretization step Δt.

3.1 *Derivation of the Stability Polynomial*

Recall that the interpolating polynomial occurring in (2.2) can be written

$$\hat{\mathbf{y}}(t_n - \omega_n) = \hat{\mathbf{y}}(t_j - \theta_n \Delta t) = \mathsf{E}^{-l}\tau(\mathsf{E}, \theta_n)\mathbf{y}_j, \qquad 0 \leqslant \theta_n < 1, \tag{3.2}$$

where $\tau(\zeta, \theta_n)$ is a polynomial of degree l in ζ with coefficients depending on θ_n. We assume (cf. Cryer, 1974) that

$$\tau(\zeta, 0) = \zeta^l, \qquad \tau(1, \theta_n) = 1. \tag{3.3}$$

Furthermore, we will always assume, in what follows, that $\omega_n \geqslant \Delta t$.

Applying the GPC method to the test equation (1.8) and writing

$$\omega = (n - j + \theta)\Delta t =: (\nu + \theta)\Delta t, \qquad z_i = \omega_i \Delta t, \tag{3.4}$$

we obtain

$$\mathbf{y}_n^{(j)} = \mu_j \mathbf{y}_n^{(j-1)} + (1 - \lambda_j - \mu_j)\mathbf{y}_n^{(j-2)} + \lambda_j b_0 (z_1 \mathbf{y}_n^{(j-1)} + z_2 \mathsf{E}^{-l}\tau(\mathsf{E}, \theta)\mathbf{y}_{n-\nu}) + \lambda_j \mathbf{w}_n. \tag{3.5}$$

Suppose that the initial approximation $\mathbf{y}_n^{(0)}$ is computed by an explicit linear multistep method $\{\tilde{\rho}, \tilde{\sigma}\}$, then by repeatedly applying (3.5) we can express $\mathbf{y}_n^{(j)}$ in terms of the step vectors $\mathbf{y}_{n-1}, \mathbf{y}_{n-2}, \mathbf{y}_{n-3}, \ldots$. In particular $\mathbf{y}_n := \mathbf{y}_n^{(m)}$ can be

expressed in terms of preceding step vectors to obtain a linear recurrence relation with constant coefficients. The corresponding characteristic polynomial or *stability polynomial* can be derived in a similar way as given in van der Houwen & Sommeijer (1983b) for the nondelay case. The result is summarized in the following theorem.

THEOREM 3.1 *Let the GPC method* (2.5) *be generated by the* $\tilde{k}$*-step predictor* $\{\tilde{\rho}, \tilde{\sigma}\}$, *the* k*-step corrector* $\{\rho, \sigma\}$, *and the* l*-step interpolation formula characterized by* τ (*cf.* (3.2)). *Then applying this method to the test equation* (1.8) *leads to the stability polynomial*

$$S_\nu(\zeta; z_1, z_2) := \zeta^{\tilde{k}+l+\nu} S(\zeta, z_1) + \gamma_m(z_1)\zeta^{k+l+\nu}\tilde{S}(\zeta, z_1) - z_2\tau(\zeta, \theta)[\zeta^{\tilde{k}}\sigma(\zeta) + \gamma_m(z_1)\zeta^k\tilde{\sigma}(\zeta)] \quad (\nu \geqslant 1), \tag{3.6}$$

where S *and* $\tilde{S}$ *are the stability polynomials of the corrector and the predictor* (*respectively given by* $S = \rho(\zeta) - z_1\sigma(\zeta)$ *and* $\tilde{S} = \tilde{\rho}(\zeta) - z_1\tilde{\sigma}(\zeta)$), *and* γ_m *is defined by*

$$\gamma_m(z_1) := (1 - b_0 z_1)\frac{P_m(z_1)}{1 - P_m(z_1)}. \tag{3.7}$$

Proof. Similar to the derivation of stability polynomials for ODEs (cf. van der Houwen & Sommeijer, 1983b), to which the result collapses on setting $z_2 = 0$. □

Evidently, the GPC method is *stable* at a point (z_1, z_2) for a given value of θ if $S_\nu(\zeta; z_1, z_2)$ is a Schur polynomial for all $\nu \in \mathbb{Z}_+$ (we will also use the terminology that $S_\nu(\zeta; z_1, z_2)$ is stable at (z_1, z_2)).

A convenient tool in the analysis of (3.6) is the familiar theorem of Rouché: *If* $f(z)$ *and* $g(z)$ *are regular on a closed region whose boundary is a closed rectifiable Jordan curve* C *and* $|g(z)| < |f(z)|$ *on* C, *then* $f(z)$ *and* $f(z)+g(z)$ *have the same number of zeros inside* C. Thus, *two polynomials* $Q(\zeta)$ *and* $R(\zeta)$ *have the same number of zeros within the unit circle if* $|R(\zeta) - Q(\zeta)| < |Q(\zeta)|$ *on the unit circle.*

Of course, this theorem provides sufficient but not necessary conditions for stability, so that the stability regions obtained may be smaller than the true stability regions (to consider a simple, if artificial, case: let $Q(\zeta) = -R(\zeta)$, then $Q(\zeta)$ and $R(\zeta)$ have common roots but the inequality given above is not satisfied). However, as we shall see in Example 3.1, in an actual situation the true stability regions are only marginally larger than what we shall call the 'Rouché-stability regions'.

3.2 *Stability of the GPC Method*

Stability plots for the GPC method employing iteration polynomials of the form

$$P_m(z) = \delta T_m\left(c + \frac{c+1}{\beta} z\right), \qquad \beta := \frac{c+1}{b_0\left[\cosh\left(\frac{1}{m}\operatorname{arccosh}\frac{1}{\delta}\right) - c\right]}, \qquad c \leqslant 1 \tag{3.8}$$

have been given in van der Houwen & Sommeijer (1983a); here, T_m denotes the Chebyshev polynomial of the first kind and δ and c are suitably chosen parame-

ters which determine the aperture and the length of stability wedge in the (z_1, z_2)-plane. The choice of the polynomials (3.8) is motivated by the large stability intervals $(-\beta, 0)$ which such polynomials generate for GPC methods without delay (cf. van der Houwen & Sommeijer, 1983b).

Since the case $|q_2| \ll |q_1|$ models an interesting class of problems associated with parabolic equations with delay, we are interested in methods with a long, narrow stability wedge along the negative z_1-axis ($z_1 = q_1 \Delta t$). Therefore, the polynomials (3.8) seem to be a good starting point for constructing efficient GPC methods. For suitable values of m and δ we refer to Section 4.1 (implementational details) where also explicit expressions for the parameters μ_j and λ_j are given.

The largest stability region, for given δ, is obtained if $c = 1$, and in this paper we only consider this case. It should be observed, however, that choosing $c = \cos(\pi/2m)$ yields an iteration polynomial which vanishes at $z = 0$ giving rise to an extra damping of the predictor error if Δt is small (see the discussion in Section 2.1). In fact, the stability plots presented in van der Houwen & Sommeijer (1983a) correspond to $c = \cos(\pi/2m)$. These plots were obtained by applying the boundary locus method to the stability polynomial S_ν defined in Theorem 3.1. A disadvantage of this direct approach is (i) we do not know *a priori* how to choose (δ, m) in order to get a stability wedge of prescribed aperture and length; (ii) we are never sure what is the effect of the delay parameter ν on the stability wedge.

In this section we propose a 'computable' approach in obtaining values for (δ, m) which more or less guarantees a stability wedge of prescribed aperture and length for all values of ν.

3.2.1 *Rouché-stability Regions* The first step is the formulation of a stability condition independent of the delay parameter ν.

THEOREM 3.2 *The GPC method* (2.4) *is stable at the point* (z_1, z_2) *if it is stable at the point* $(z_1, 0)$ *and if*

$$|z_2| < A_m(z_1, \theta) := \inf_{|\zeta|=1} \left| \frac{S(\zeta, z_1) + \gamma_m(z_1)\zeta^{k-\tilde{k}}\tilde{S}(\zeta, z_1)}{\tau(\zeta, \theta)[\sigma(\zeta) + \gamma_m(z_1)\zeta^{k-\tilde{k}}\tilde{\sigma}(\zeta)]} \right|.$$

Proof. Applying Rouché's theorem with

$$Q(\zeta) = \zeta^{l+\nu}[S(\zeta, z_1)\zeta^{\tilde{k}} + \gamma_m(z_1)\tilde{S}(\zeta, z_1)\zeta^k], \qquad R(\zeta) = S_\nu(\zeta; z_1, z_2)$$

the theorem follows immediately. □

In order to obtain a region of stable points (z_1, z_2) we shall determine the stability interval on the z_1-axis for the GPC method, that is the stability interval in the case of a vanishing delay. This special case was studied in van der Houwen & Sommeijer (1983b). For a GPC method generated by an extrapolation predictor and a backward differentiation corrector (EP-BD pair), δ-values for the polynomial $P_m(z)$ were derived such that the GPC method is stable in the interval $-\beta < z_1 < 0$ with β defined in (3.8) (we denote stable δ-values for the nondelay case by δ^*). For future reference these values are listed in Table 1.

TABLE 1
Stable δ^*-values for GPC methods without delay generated by $EP_{\tilde{k}}-BD_k$ pairs

k	$\tilde{k}=1$	$\tilde{k}=2$	$\tilde{k}=3$	$\tilde{k}=4$	$\tilde{k}=5$	$\tilde{k}=6$	$\tilde{k}=7$
2	1	1/3	1/7				
3	1	1/3	1/7	1/15			
4	0·75	1/3	1/7	1/15	1/31		
5	0·44	0·33	1/7	1/15	1/31	1/63	
6	0·13	0·07	0·07	1/15	0·0289	0·0147	1/127

Using larger δ-values gives rise to the development of instabilities but not in a severe way. In the numerical experiments reported in Section 4, it will be demonstrated that using larger values will still produce useful results. The reason is that the stability polynomial does not rapidly increase in magnitude if δ increases beyond the nondelay value δ^*. (In contrast, violating the stability condition $z_1 = q_1\Delta t > -\beta$, i.e. $\Delta t < \beta/|q_1|$ leads to a rapid increase of the magnitude of $P_m(z)$ for $z < -\beta$.) Therefore, the values of δ^* listed in Table 1 should be used as an *indication of the acceptable upper bound for* δ *in using EP-BD methods*; in actual computation, one may often use much larger values.

In the following example we give the values of the 'effective' length $L(\delta, m)/m$ and the aperture angle $\psi(\delta, m)$ of the wedge contained in the stability region for a fourth-order EP-BD method (see Fig. 2). We limit ourselves by, from all wedges contained in the stability region, the one with maximal aperture 2ψ. The factor $1/m$ is applied to $L(\delta, m)$ because $O(m)$ operations are employed within each step.

EXAMPLE 3.1 By virtue of Theorem 3.2 we can compute estimates of the stability wedge $\{L(\delta, m)/m, \psi(\delta, m)\}$ for any given predictor-corrector pair. For the pair EP_5-BD_4 some results are listed in Table 2; the value of c occurring in (3.8) was set to 1 and $\tau(\zeta, \theta)$ was chosen of degree $l = 4$. The values of L are slightly smaller than $\beta(\delta, m)$ given by (3.8). To get some idea about the pessimism due to

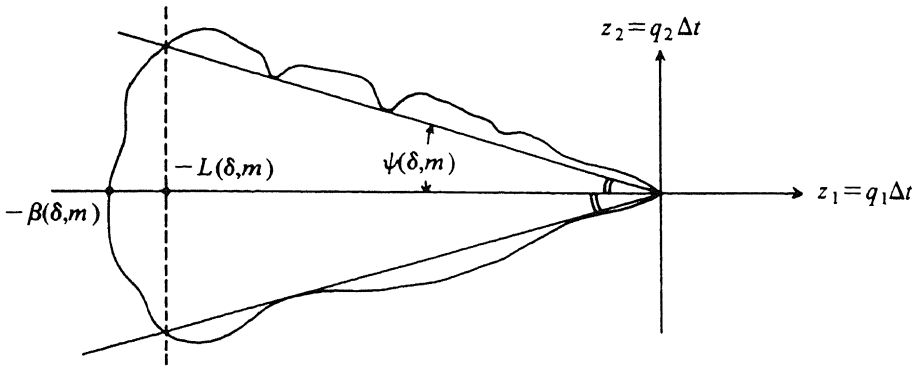


FIG. 2. Stability wedge defined by $L(\delta, m)$ and $\psi(\delta, m)$.

TABLE 2
EP_5-BD_4 *method*: $(L/m, \psi)$-*values derived from Theorem* 3.2 (ψ *is given in degrees*)

δ	$\theta=0$ $m=2$	$m=8$	$m=64$	$\theta=0{\cdot}5$ $m=2$	$m=8$	$m=64$
0·01	(0·34,45)	(2·3,42)	(19,35)	(0·34,45)	(2·2,35)	(19,23)
$\delta^*=1/31$	(0·69,45)	(3·8,16)	(31,0·25)	(0·69,45)	(3·8,10)	(31,0·16)
0·10	(1·4,40)	(7·3,0·12)	(59,0·09)	(1·4,29)	(7·3,0·08)	(59,0·06)

the estimates obtained through use of Rouché's Theorem we calculated in addition the true stability regions for the parameters given in Table 2. Since these regions depend on the value of ν (cf. (3.6)), we made plots for several ν-values and determined the length and aperture of the stability wedge contained in all these stability regions. It turned out that these values (i) hardly depend on the value of ν and (ii) *are only slightly larger* than those listed in Table 2.

3.2.2 *Choice of the Predictor–corrector Pair* It is of interest to observe that we are more or less forced to use *extrapolation predictors* if large β-values are desired. To see this we apply Rouché's theorem to the polynomial $S_\nu(\zeta; z_1, 0)$ with $Q(\zeta)=S(\zeta, z_1)$ and $R(\zeta)=S_\nu(\zeta; z_1, 0)$ to obtain the stability condition

$$|\gamma_m(z_1)|<\inf_{|\zeta|=1}\left|\frac{S(\zeta, z_1)}{\tilde{S}(\zeta, z_1)}\right|=\inf_{|\zeta|=1}\left|\frac{\rho(\zeta)-z_1\sigma(\zeta)}{\tilde{\rho}(\zeta)-z_1\tilde{\sigma}(\zeta)}\right|.$$

Since $|\gamma_m(z_1)|$ is proportional to $|z_1|$ for $|z_1|$ large we should choose $\tilde{\sigma}\equiv 0$ in order to get a large stability interval $(-\beta, 0)$. Predictor formulas with a vanishing polynomial $\tilde{\sigma}$ are just the extrapolation predictors characterized by $\tilde{\rho}(\zeta)=(\zeta-1)^k$.

In order to choose a suitable corrector $\{\rho, \sigma\}$ we consider the stability wedge (β, ψ) as $\delta\to 0$. (Notice that $\delta=0$ implies $P_m(z)\equiv 0$ (cf. (3.8)), that is the delay–corrector equation is iterated to convergence.) From Theorem 3.2 the following corollary follows:

COROLLARY 3.1. *Let the corrector formula* $\{\rho, \sigma\}$ *be* $A(\alpha)$*-stable. Then the delay-corrector formula* (2.3) *is* $P_0(\psi)$*-stable and*

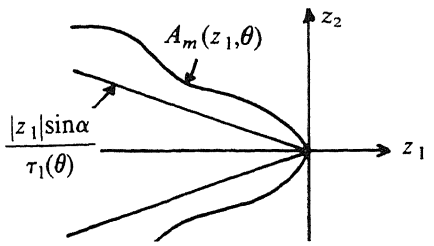
$$\psi\geqslant\arctan\frac{\sin\alpha}{\tau_1(\theta)}, \qquad \tau_1(\theta):=\sup_{|\zeta|=1}|\tau(\zeta, \theta)|. \tag{3.9}$$

Proof. We have to show that the stability region of the GPC method with $\delta=0$ (i.e. $\gamma_m(z)\equiv 0$) contains the real infinite wedge $|z_2/z_1|<\tan\psi$ where ψ satisfies (3.9). From Theorem 3.2 with $\delta=0$ and by virtue of the $A(\alpha)$-stability of the generating $\{\rho, \sigma\}$ formula we derive the stability region

$$|z_2|<A_m(z_1, \theta)=\inf_{|\zeta|=1}\left|\frac{1}{\tau(\zeta, \theta)}\left(\frac{\rho}{\sigma}(\zeta)-z_1\right)\right|, \qquad z_1\leqslant 0.$$

Furthermore, it is easily verified that an $A(\alpha)$-stable LM formula satisfies

$$\inf_{|\zeta|=1}\left|\frac{\rho}{\sigma}(\zeta)-z\right|\geqslant|z|\sin\alpha$$



$$|z_2| < \frac{|z_1|\sin\alpha}{\tau_1(\theta)}, \quad z_1 < 0$$

FIG. 3.

for all $z \leqslant 0$. Thus,

$$A_m(z_1, \theta) \geqslant \frac{1}{\tau_1(\theta)} \inf_{|\zeta|=1} \left|\frac{\rho}{\sigma}(\zeta) - z_1\right| \geqslant \frac{|z_1| \sin \alpha}{\tau_1(\theta)}.$$

The stability region of the delay–corrector formula therefore (see Fig. 3) satisfies $|z_2| < (|z_1| \sin \alpha)/\tau_1(\theta)$ with $z_1 < 0$, and certainly contains the infinite wedge $|z_2/z_1| < \tan \psi$ for all ψ satisfying (3.9). □

This corollary shows the strong relation between $P_0(\psi)$-stability of the delay–corrector equation and the $A(\alpha)$-stability of the generating LM method. For parabolic equations with delay (excluding delay equations of type (1.3)), we only need stability wedges with relatively small apertures ψ; therefore we can limit our considerations to generating LM methods $\{\rho, \sigma\}$ that are $A(\alpha)$-stable where α is allowed to be small. Such LM methods are provided by the backward differentiation formulae for which α varies from 90° for $k = 2$ to 18° for $k = 6$ (cf. Lambert, 1973).

A second conclusion from Corollary 3.1 is the strong dependency of ψ on $\tau(\zeta, \theta)$. The following example presents values of the lower bound on ψ for various values of θ.

EXAMPLE 3.2. Consider the delay–corrector formulae generated by the backward differentiation formulas BD_k and by interpolation polynomials τ of degree $l = k$. Then the lower bounds in (3.9) are given in Table 3 for a few values of θ. From this table we see that it is advantageous to select the step size in such a way that $t_n - \omega_n$ coincides with a step point (i.e. $\theta = 0$); however, if interpolation is performed, at least $\frac{2}{3}$ of the maximal ψ-value is obtained.

TABLE 3
BD_k *methods*: ψ *lower bounds derived from Corollary 3.1* (ψ *is given in degrees*)

θ	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0	45·0	45·0	43·7	37·9	17·2
0·25	39·2	39·2	37·9	32·3	14·1
0·5	31·6	31·6	30·5	25·6	10·8
0·75	33·0	33·0	31·9	26·8	11·4
1	45·0	45·0	43·7	37·9	17·2

It would be convenient to have the analogue of Corollary 3.1 for the GPC method itself. It is not difficult to find such an analogue for large $|z_1|$ and small values of δ; however, if δ increases, the resulting lower bounds become rather pessimistic. Therefore, we also considered upper bounds for ψ. The following corollary of Theorem 3.2 presents these results.

COROLLARY 3.2 *Let δ_0 be the maximal stable δ-value of the GPC method for nondelay equations, let $\beta = \beta(\delta, m)$ be defined by (3.8) and let $\{\rho, \sigma\}$ be $A(\alpha)$-stable. Then for $-\beta(\delta_0, m) \leq z_1 \ll -1$ the stability region of the GPC method employing an extrapolation predictor $EP_{\bar{k}}$ is bounded by $|z_2/z_1| < \tan\psi$ where $\psi = \psi(\delta, m)$ satisfies the inequality*

$$\frac{1}{\tau_1(\theta)}\left(\sin\alpha - 2^{\bar{k}}\frac{b_0\delta}{1-\delta}\sup_{|\zeta|=1}\frac{1}{|\sigma(\zeta)|}\right) \leq \tan\psi \leq \frac{1}{|\tau(-1,\theta)|}\left(1 - 2^{\bar{k}}\frac{b_0\delta}{1-\delta}\frac{1}{|\sigma(-1)|}\right) \tag{3.10}$$

with $\tau_1(\theta)$ defined in (3.9), provided that δ is sufficiently small and that the right- and left-hand side in (3.10) are positive.

Proof. For extrapolation predictors we have $\tilde{\rho}(\zeta) = (\zeta - 1)^{\bar{k}}$ and $\tilde{\sigma}(\zeta) = 0$ so that the function $A_m(z_1, \theta)$ can be written as

$$A_m(z_1, \theta) = \inf_{|\zeta|=1}\left|\frac{1}{\tau(\zeta, \theta)}\left(\frac{\rho}{\sigma}(\zeta) - z_1 + \gamma_m(z_1)\zeta^{k-\bar{k}}\frac{(\zeta-1)^{\bar{k}}}{\sigma(\zeta)}\right)\right|. \tag{3.11}$$

First we derive the lower bound for ψ. Similar to the derivation of the lower bound (3.9) we obtain

$$\begin{aligned} A_m(z_1, \theta) &\geq \frac{1}{\tau_1(\theta)}\inf_{|\zeta|=1}\left(\left|\frac{\rho}{\sigma}(\zeta) - z_1\right| - \left|\gamma_m(z_1)\frac{(\zeta-1)^{\bar{k}}}{\sigma(\zeta)}\right|\right) \\ &\geq \frac{1}{\tau_1(\theta)}\left(|z_1|\sin\alpha - 2^{\bar{k}}|\gamma_m(z_1)|\sup_{|\zeta|=1}\frac{1}{|\sigma(\zeta)|}\right). \end{aligned}$$

The stability region therefore satisfies

$$|z_2| < \frac{|z_1|}{\tau_1(\theta)}\left(\sin\alpha - 2^k\left|\frac{\gamma_m(z_1)}{z_1}\right|\sup_{|\zeta|=1}\frac{1}{|\sigma(\zeta)|}\right)$$

provided that the right-hand side is positive, that is, $|\gamma_m/z_1|$ is sufficiently small; this is achieved by choosing δ sufficiently small as can be seen from the result

$$\left|\frac{\gamma_m(z_1)}{z_1}\right| = \left|\frac{1}{z_1} - b_0\right|\left|\frac{P_m(z_1)}{1-P_m(z_1)}\right| \leq \left|\frac{1}{z_1} - b_0\right|\frac{\delta}{1-\delta}.$$

Using this upper bound on $|\gamma_m/z_1|$ and assuming $|z_1|$ large we arrive at the left-hand inequality in (3.10).

Next we derive the upper bound on ψ. From (3.11) it follows that for $|z_1| \ll 1$

$$\begin{aligned} A_m(z_1, \theta) &\leq \frac{1}{\tau(-1,\theta)}\left|\frac{\rho}{\sigma}(-1) - z_1 + \gamma_m(z_1)\frac{(-1)^k 2^{\bar{k}}}{\sigma(-1)}\right| \\ &\approx \frac{|z_1|}{|\tau(-1,\theta)|}\left|1 - \frac{\gamma_m(z_1)}{z_1}\frac{(-1)^k 2^{\bar{k}}}{\sigma(-1)}\right|. \end{aligned}$$

Thus, for $|\gamma_m/z_1|$ sufficiently small the stability region satisfies

$$|z_2| \leqslant \frac{|z_1|}{|\tau(-1,\theta)|}\left(1-\left|\frac{\gamma_m(z_1)}{z_1}\right|\frac{2^{\tilde{k}}}{|\sigma(-1)|}\right)$$
$$\leqslant \frac{|z_1|}{|\tau(-1,\theta)|}\left(1-\frac{b_0\delta}{1-\delta}\frac{2^{\tilde{k}}}{|\sigma(-1)|}\right)$$

which leads to the right-hand inequality in (3.10). □

EXAMPLE 3.3 For $\mathrm{EP}_{\tilde{k}}-\mathrm{BD}_k$ methods Corollary 3.2 yields

$$\frac{1}{\tau_1(\theta)}\left(\sin\alpha - 2^{\tilde{k}}\frac{\delta}{1-\delta}\right) \leqslant \tan\psi \leqslant \frac{1}{|\tau(-1,\theta)|}\left(1-2^{\tilde{k}}\frac{\delta}{1-\delta}\right), \tag{3.12}$$

where it is assumed that δ is sufficiently small to provide positive right- and left-hand sides. Evidently, δ should satisfy the inequality

$$\delta \leqslant \sin\alpha/(2^{\tilde{k}}-\sin\alpha).$$

For the predictor–corrector pair mentioned in Example 3.1, we find for $\delta = 0{\cdot}01$, and respectively $\theta = 0$ and $\theta = 0{\cdot}5$ the ψ-ranges $32{\cdot}3° \leqslant \psi \leqslant 34{\cdot}1°$ and $21.3° \leqslant \psi \leqslant 22.6°$. For large values of m, these ψ-bounds are in good agreement with the ψ-values given in Table 2. (We emphasize that (3.12) has been derived under the assumption that $z_1 \ll -1$.)

4. Numerical illustrations

A most important aspect of the numerical integration of parabolic equations with delay is the storage requirements. As any algorithm needs an (interpolated) approximation of the delay term, we have to store at least ν arrays of $\mathbf{y}$-vectors (notice that $\nu \approx \omega_n/\Delta t$ may change from step to step). Moreover, the dimension of these $\mathbf{y}$-vectors is usually very large and their storage requires a tremendous computer memory capacity. Therefore, in order to reduce the value of ν, it is of vital importance to be able to integrate with large time steps. However, the use of large time steps demands good stability properties. Of course, one possibility is to select an implicit method; for an extensive survey of such methods we refer to Cryer (1972) (see also Tavernini, 1973). However, implicit methods require, in each time step, the solution of large systems of equations. Apart from the computational effort involved in solving these systems, this aspect implies again that a considerable amount of storage is required, especially in the case of *higher*-dimensional parabolic equations.

An alternative to circumvent this huge algebraic problem is the use of splitting methods such as ADI (for a description, see below) in which case the Jacobians usually possess a tridiagonal structure; hence, storage requirements are modest. However, a disadvantage of this type of method is the low order of accuracy. Moreover, an important aspect of using large time steps in fully implicit methods as well as in partially implicit (splitting) methods, is the problem of constructing a sufficiently accurate initial approximation in order to let the Newton process be convergent when solving nonlinear problems. For a detailed discussion concerning the implementational aspects of implicit methods we refer to Lambert (1973).

Taking all these considerations into account, we think that the EP-BD methods, by which a high order, good stability behaviour, and minimal storage requirements are combined, are a useful tool for integrating parabolic equations with delay.

To illustrate its performance we will separately test two different features of the GPC method, viz. its accuracy (or efficiency) and its stability.

(i) The accuracy is shown in Sections 4.2 and 4.3; here we give results obtained by $EP_p - BD_p$ methods for several values of p as well as the results obtained by the ADI method. Fully implicit methods are not implemented because of their enormous storage requirements.

The test examples are constructed by choosing an exact solution, a (nonlinear) differential operator and a delay term. As these terms usually do not match, we have to add an inhomogeneous term. However, the above terms are chosen in such a way as to avoid a dominant influence of this inhomogeneous term in the whole interval of integration. This procedure is motivated by our wish to have a solution which does not converge to a steady state as is the tendency of solutions of parabolic (delay) equations without a source term. Both test examples have an initial ϕ-function (cf. (1.2)) which coincides with the exact solution, hence no discontinuities in higher derivatives of the solution will occur. This enables us to use the exact solution, which is convenient for measuring the accuracies. Moreover, since in these accuracy tests the emphasis is on *time*-integration aspects, we took care that the space-dependent part of the solution has a smooth behaviour; more precisely, they are chosen in such a way that the space discretization does not introduce an error, i.e. the solution of the system of ODEs with delay equals the solution of the PDE, restricted to the grid points. This spatial discretization is achieved using standard 5-point molecules on a uniform mesh with mesh size $h = \frac{1}{20}$. To measure the accuracy of the various methods we define

$$a_{cd} := -\log_{10} \|\text{absolute error at the endpoint}\|_\infty, \tag{4.1}$$

denoting the number of correct decimals in the answer.

(ii) The stability behaviour of the GPC methods is illustrated in Section 4.4, where we choose an initial Φ-function which is wildly varying both in space and time.

4.1 *Implementational Details*

4.1.1 *EP-BD Schemes* In constructing an EP-BD scheme our starting point is the iteration polynomial $P_m(z)$ as given in (3.8). By choosing $c = 1$ the length of the stability wedge is maximized but, as a consequence, $P_m(z)$ does not vanish at $z = 0$. Therefore, we combined a p-step BD corrector with a $(p+1)$-step EP predictor (which are both of order p), resulting in a pth order EP-BD method (see Theorem 2.1 and the discussion thereafter). The choice $c = 1$ yields

$$\beta = \frac{2/b_0}{\cosh\left(\frac{1}{m}\operatorname{arccosh}\frac{1}{\delta}\right) - 1}. \tag{4.2}$$

Now, m should be chosen sufficiently large to satisfy the stability condition $|q_1|\,\Delta t<\beta$. Here, $|q_1|$ stands for the spectral radius of the Jacobian matrix and will depend on the problem (see also Sections 4.2 and 4.3). It should be observed that this condition is a condition on m rather than a condition on the time step; consequently, Δt may be chosen merely on the basis of accuracy considerations. In the actual application of these methods the values of m and $1/\delta$ will be large; therefore the following asymptotic expression will be useful:

$$\beta \sim \frac{4m^2/b_0}{[\ln(2/\delta)]^2} \quad (m\to\infty,\ \delta\ll 1).$$

Note that the stability boundary β shows an $O(m^2)$ behaviour.

Once we have determined an m-value that ensures stability, we will derive expressions for the parameters μ_j and λ_j. Let us define the iteration polynomials

$$P_j(z)=\delta_j T_j\left(1+\frac{2}{\beta}z\right) \quad (j=0,\ldots,m-1), \tag{4.3}$$

where $\delta_j=1/T_j(1+2/b_0\beta)$ in order to satisfy $P_j(1/b_0)=1$ (cf. (2.6)). By virtue of the well known three-term Chebyshev recursion the polynomials P_j satisfy

$$\begin{aligned} &P_0(z)=1, \qquad P_1(z)=\left(1+\frac{2z}{\beta}\right)\delta_1,\\ &P_j(z)=2\left(1+\frac{2z}{\beta}\right)\frac{\delta_j}{\delta_{j-1}}P_{j-1}(z)-\frac{\delta_j}{\delta_{j-2}}P_{j-2}(z) \quad (j=2,\ldots,m), \end{aligned} \tag{4.4}$$

where we have set $\delta_m=\delta$. Now, identification of (2.6) and (4.4) yields the parameters of the scheme (2.5)

$$\begin{aligned} &\mu_1=1-\lambda_1, \qquad \lambda_1=\frac{2}{b_0\beta}\delta_1,\\ &\mu_j=2\frac{\delta_j}{\delta_{j-1}}, \qquad \lambda_j=\frac{4}{b_0\beta}\frac{\delta_j}{\delta_{j-1}} \quad (j=2,\ldots,m). \end{aligned} \tag{4.5}$$

4.1.2 *Nonlinear ADI Scheme* A well-known method for parabolic equations without delay is the ADI method of Peaceman and Rachford (Peaceman & Rachford, 1955 and van der Houwen & Verwer, 1979). As this method combines modest storage requirements with good stability properties we implemented, for the sake of comparison, an adjusted form of this method to make it suitable to integrate delay equations. The ADI method requires a splitting of the function $\boldsymbol{f}(t, \mathbf{y}(t), \mathbf{y}(t-\omega))$ in (2.1). We assume that $\boldsymbol{f}$ can be written as $\boldsymbol{f}(t, \mathbf{y}(t), \mathbf{y}(t-\omega))=\boldsymbol{f}_1(t, \mathbf{y}(t), \mathbf{y}(t-\omega))+\boldsymbol{f}_2(t, \mathbf{y}(t), \mathbf{y}(t-\omega))$, where the functions $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ correspond to the one-dimensional differential operators in the x_1 and x_2 directions, respectively. Now, the (nonlinear) ADI method is defined by

$$\begin{aligned} &\mathbf{y}^*=\mathbf{y}_{n-1}+\tfrac{1}{2}\Delta t\boldsymbol{f}_1(t_{n-1}+\tfrac{1}{2}\Delta t, \mathbf{y}^*, \hat{\mathbf{y}}_{n-\frac{1}{2}})+\tfrac{1}{2}\Delta t\boldsymbol{f}_2(t_{n-1}, \mathbf{y}_{n-1}, \hat{\mathbf{y}}(t_{n-1}-\omega_{n-1})),\\ &\mathbf{y}_n=2\mathbf{y}^*-\mathbf{y}_{n-1}+\tfrac{1}{2}\Delta t\boldsymbol{f}_2(t_n, \mathbf{y}_n, \hat{\mathbf{y}}(t_n-\omega_n))-\tfrac{1}{2}\Delta t\boldsymbol{f}_2(t_{n-1}, \mathbf{y}_{n-1}, \hat{\mathbf{y}}(t_{n-1}-\omega_{n-1})), \end{aligned} \tag{4.6}$$

where

$$\hat{y}_{n-\frac{1}{2}} = \hat{y}(t_{n-1} + \tfrac{1}{2}\Delta t - \omega(t_{n-1} + \tfrac{1}{2}\Delta t)).$$

The inhomogeneous term, if any, is equally distributed over f_1 and f_2.

4.2 *A Mildly Nonlinear Example*

As a first example consider the parabolic equation, defined on the unit square in the (x_1, x_2)-plane

$$\left.\begin{aligned} \frac{\partial}{\partial t} u(t, x_1, x_2) = \tfrac{1}{3} \frac{1 + x_1 + x_2}{1 + t} \left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} \right) u^3(t, x_1, x_2) - 4 \frac{u^3(t-1, x_1, x_2)}{1+t} + \\ \tfrac{2}{3}\pi(1 + x_1 + x_2) \cos 2\pi t \quad (0 \leqslant t \leqslant 2), \\ \phi(t, x_1, x_2) = \tfrac{1}{3}(1 + x_1 + x_2) \sin 2\pi t \quad (t \leqslant 0). \end{aligned}\right\} \tag{4.7}$$

The solution $u(t, x_1, x_2)$ equals the function ϕ for all t. The Dirichlet boundary conditions are taken from the solution u.

In order to get a stability wedge of sufficient length, that is to have an m-value which is sufficiently large, we must have an estimate of the spectral radius B of the Jacobian matrix $\partial \mathbf{f}/\partial \mathbf{y}$. We used

$$B_{n-1} = [B(\partial \mathbf{f}/\partial \mathbf{y})]_{t=t_{n-1}} = 1{\cdot}1 \frac{72}{h^2} \max_{t \in [t_{n-1}, t_n]} \frac{\sin^2 2\pi t}{1+t},$$

where the factor $1{\cdot}1$ is added to obtain a safe upper-bound.

In Tables 4, 5, and 6 we give the results of the second-, fourth-, and sixth-order

TABLE 4

(a_{cd}/N)-values for the second-order EP-BD *method; the total number of arrays equals* $1/\Delta t + 4$

δ	$\Delta t = 1/10$	$\Delta t = 1/20$	$\Delta t = 1/40$
0·1	1·4/824	1·8/1059	2·4/1418
$\delta^* = 1/7$	1·4/725	1·8/935	2·5/1256
0·2	1·2/632	1·8/819	2·5/1094
0·4	*	*	2·2/759

TABLE 5

(a_{cd}/N)-values for the fourth-order EP-BD *method; the total number of arrays equals* $1/\Delta t + 4$

δ	$\Delta t = 1/10$	$\Delta t = 1/20$	$\Delta t = 1/40$
0·01	2·0/1231	3·2/1585	4·3/2115
$\delta^* = 1/31$	1·9/960	3·2/1238	4·3/1658
0·1	1·6/700	3·0/903	4·2/1217
0·15	*	1·8/779	2·6/1053

TABLE 6
(a_{cd}/N)-values for the sixth-order EP-BD *method; the total number of arrays equals* $1/\Delta t+4$

δ	$\Delta t = 1/10$	$\Delta t = 1/20$	$\Delta t = 1/40$
0·005	2·2/1284	4·4/1649	6·1/2202
$\delta^* = 1/127$	2·2/1186	4·4/1527	6·1/2039
0·02	2·0/989	4·4/1275	6·1/1706
0·04	1·2/842	3·6/1087	5·5/1458

TABLE 7
a_{cd}-values for the ADI method; the total number of arrays equals $1/\Delta t+10$ *(including both tridiagonal Jacobian matrices). The total number of iterations to solve the implicit relations equals* $4\times$NEWT$/\Delta t$.

NEWT	$\Delta t = 1/40$	$\Delta t = 1/60$	$\Delta t = 1/80$	$\Delta t = 1/120$
1	*	*	1·9	2·2
2	*	2·4	2·8	3·5
5	*	3·9	4·2	4·5

EP-BD method, respectively, for several values of the time step Δt. In these tables a_{cd} is defined in (4.1) and N denotes the total number of iterations, summed over all time steps. Note that the number of iterations per time step is not constant because the spectral radius B_n varies in time. An '*' denotes unstable behaviour. A mutual comparison of the EP-BD methods reveals that the higher-order formulae are the more efficient ones.

Furthermore, concerning the value of δ we see that a larger value is allowed than indicated by Table 1, but the methods gradually lose accuracy as δ increases. This is due to a mild form of instability.

The results obtained by the second-order ADI method are listed in Table 7. We applied the method for several values of NEWT, being the number of Newton iterations performed to 'solve' each implicit relation in (4.6). Note that for the EP-BD methods an iteration is simply an f-evaluation; for the ADI method, however, an iteration is of quite a different nature and usually much more expensive (i.e. one evaluation of f and the solution of a tridiagonal system of equations). Moreover, the Jacobian matrices have to be evaluated (in this experiment we updated the Jacobians every step). Hence, a comparison of both methods in terms of efficiency is not feasible.

However, as the ADI method needs a relatively small time step for stability reasons, its storage requirements tend to become excessive, whereas the EP-BD methods can take rather large time steps, thus reducing the number of y-vectors to be kept in store.

4.3 *A Strongly Nonlinear Example*

To construct our second test problem we employ the 'porous-medium operator'

$$\Delta(u(t, x_1, x_2))^\kappa, \qquad \kappa \geq 2,$$

and specify the analytic solution as

$$u(t, x_1, x_2) = \tfrac{1}{4}(x_1 + x_2)^{2/\kappa}[e^{-2(t-1)^2} + e^{-2(t-3)^2} + 1]. \tag{4.8}$$

The initial function $\phi(t, x_1, x_2)$ and the Dirichlet boundary conditions are prescribed by (4.8). By setting $\kappa = 5$ and introducing a delay term and an inhomogeneous term $g(t, x_1, x_2)$ we arrive at

$$\left.\begin{aligned} \frac{\partial}{\partial t} u(t, x_1, x_2) &= \left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}\right) u^5(t, x_1, x_2) + 4u(t-2, x_1, x_2) - 4u(t, x_1, x_2) + \\ &\qquad g(t, x_1, x_2) \quad (1 \leqslant t \leqslant 7), \\ \phi(t, x_1, x_2) &= \tfrac{1}{4}(x_1 + x_2)^{\frac{2}{5}}[e^{-2(t-1)^2} + e^{-2(t-3)^2} + 1] \quad (t \leqslant 1) \end{aligned}\right\} \tag{4.9}$$

defined on the unit square in the (x_1, x_2)-plane.

For this problem, a safe upper bound for the spectral radius B of $\partial \mathbf{f}/\partial \mathbf{y}$ is obtained by

$$[B(\partial \mathbf{f}/\partial \mathbf{y})]_{t=t_{n-1}} \approx 1 \cdot 1 \, \frac{120}{h^2} \frac{1}{4^4} \max_{t \in [t_{n-1}, t_n]} (e^{-2(t-1)^2} + e^{-2(t-3)^3} + 1)^4.$$

Similar to the previous example we tested the second-, fourth-, and sixth-order EP-BD method as well as the ADI method. The results are given in the Tables 8–11.

Again, an * denotes unstable behaviour of the integration process and the quantities a_{cd}, N, and NEWT have the same meaning as defined in Section 4.2. The results of this example give rise to conclusions similar to those of the previous example: to obtain a stable result, the ADI method needs a smaller time step Δt than the GPC method does. Again, the higher order of the GPC methods are more efficient unless only moderate accuracy is required ($a_{cd} \leqslant 2.5$, say).

TABLE 8

(a_{cd}/N)-values for the second-order EP-BD *method; the total number of arrays equals* $2/\Delta t + 4$

δ	$\Delta t = 1/4$	$\Delta t = 1/8$	$\Delta t = 1/16$
0·1	2·3/436	2·7/610	3·4/852
$\delta^* = 1/7$	2·2/382	2·7/532	3·5/739
0·2	2·2/332	2·7/463	3·6/665
0·3	1·4/278	1·5/381	1·5/539

TABLE 9

(a_{cd}/N)-values for the fourth-order EP-BD *method; the total number of arrays equals* $2/\Delta t + 4$

δ	$\Delta t = 1/4$	$\Delta t = 1/8$	$\Delta t = 1/16$
0·01	2·7/652	4·1/895	5·1/1247
$\delta^* = 1/31$	2·7/513	4·0/704	5·1/1000
0·1	*	1·9/517	5·1/716

TABLE 10
(a_{cd}/N)-values for the sixth-order EP-BD *method; the total number of arrays equals* $2/\Delta t+4$

δ	$\Delta t=1/4$	$\Delta t=1/8$	$\Delta t=1/16$
0·005	2·0/677	4·7/924	7·1/1327
$\delta^*1/127$	2·0/625	4·7/864	7·1/1214
0·02	2·0/522	4·7/720	7·0/1017
0·03	2·0/478	4·1/654	4·7/920

TABLE 11
a_{cd}-values for the ADI method; the total number of arrays equals $2/\Delta t+10$; *the total number of Newton iterations equals* $12\times\text{NEWT}/\Delta t$

NEWT	$\Delta t=1/4$	$\Delta t=1/8$	$\Delta t=1/16$	$\Delta t=1/32$
1	*	*	2·6	3·0
2	*	3·1	3·8	4·4
5	2·8	3·6	4·3	5·0

4.4 *Stability Test*

To perform a stability test we employ the same differential operator as discussed in the previous section but now an initial ϕ-function is closed which is wildly varying both in space and time:

$$\left.\begin{aligned}\frac{\partial}{\partial t}u(t,x_1,x_2)&=\left(\frac{\partial^2}{\partial x_1^2}+\frac{\partial^2}{\partial x_2^2}\right)u^5(t,x_1,x_2)+4u(t-2,x_1,x_2)\quad(1\leqslant t\leqslant 7),\\ \phi(t,x_1,x_2)&=\tfrac{1}{10}e^{x_1^2+x_2^3}e^{-2(t-1)^2}\quad(t\leqslant 1).\end{aligned}\right\}\tag{4.10}$$

As a consequence of this choice, we have no analytical solution available.

To obtain insight into the robustness of the methods with respect to stability we perform the following test: first we semidiscretize (4.10) where the Dirichlet boundary conditions are taken from ϕ if $t\leqslant 1$ and are fixed in time for $t>1$ and given by $\phi(1,\bullet,\bullet)$. Now, the resulting system of ODEs is integrated in time, resulting in a numerical solution, say, v_1 at $t=7$. Next, a solution v_2 is determined by solving the same equation but now the function ϕ is (relatively) disturbed by an amount $\text{rn}\times 10^{-2}$, where rn is randomly chosen from $(-1,1)$ and being different for each component of the system and for each value of t.

At $t=7$ we compare both numerical solutions to see to what extent the methods have damped or amplified the initial perturbations; to measure the amplification factors we define

$$F:=10^2\|v_1-v_2\|_\infty.\tag{4.11}$$

In Tables 12–15 we tabulated these factors for the GPC methods of orders 2, 4, and 6 and for the ADI method, respectively. As can be seen from these tables the robustness of the GPC methods decreases as the order increases, a phenomenon which is commonly encountered in using linear multistep methods. However, in

TABLE 12
F-values for the second-order EP-BD *method*

δ	$\Delta t = 1/2$	$\Delta t = 1/4$	$\Delta t = 1/8$	$\Delta t = 1/16$
0·1	$2{\cdot}3_{10}-3$	$3{\cdot}3_{10}-4$	$6{\cdot}5_{10}-4$	$3{\cdot}2_{10}-4$
$\delta^* = \frac{1}{7}$	0·4	$3{\cdot}9_{10}-4$	$7{\cdot}5_{10}-4$	$3{\cdot}2_{10}-4$
0·2	0·4	$3{\cdot}4_{10}-3$	$4{\cdot}1_{10}-2$	14·9
0·3	*	*	*	*

TABLE 13
F-values for the fourth-order EP-BD *method*

δ	$\Delta t = 1/4$	$\Delta t = 1/8$	$\Delta t = 1/16$
0·01	*	$1{\cdot}0_{10}-3$	$2{\cdot}3_{10}-4$
$\delta^* = 1/31$	*	$1{\cdot}1_{10}-3$	$1{\cdot}8_{10}-4$
0·05	*	$1{\cdot}5_{10}-2$	1·1
0·1	*	*	*

TABLE 14
F-values for the sixth-order EP-BD *method*

δ	$\Delta t = 1/8$	$\Delta t = 1/16$	$\Delta t = 1/32$
0·005	*	$5{\cdot}8_{10}-4$	$4{\cdot}4_{10}-4$
$\delta^* = 1/127$	*	$8{\cdot}9_{10}-4$	$4{\cdot}5_{10}-4$
0·01	*	$1{\cdot}3_{10}-3$	*
0·02	*	*	*

TABLE 15
F-values for the ADI method

NEWT	$\Delta t = 1/16$	$\Delta t = 1/32$
1	*	*
2	*	$6{\cdot}0_{10}-5$
5	*	$6{\cdot}0_{10}-5$

comparison with the ADI method, all GPC methods tested can deal with a larger time step as far as stability is concerned, thus reducing the number of vectors to be stored.

5. Conclusion

We have indicated how a class of methods for certain parabolic equations with delay can be derived by extending the GPC methods for semidiscretized parabolic

equations. The resulting methods have the following properties:

(i) The GPC method consists of an (explicit) linear multistep predictor, an (implicit) linear multistep corrector and an (unconventional) iteration scheme.

(ii) In order to relax the stability conditions for this method the predictor should be based on extrapolation of preceding $\mathbf{y}_n$-values and the corrector should be $A(\alpha)$-stable where α is allowed to be relatively small.

(iii) The integration step may be freely chosen because the number of iterations is automatically adapted to ensure stability; therefore, the integration step is determined only by accuracy considerations and not limited by stability.

(iv) By choosing a high-order corrector the method can take large integration steps (as far as accuracy is concerned) thereby limiting the number of back values which should be stored to compute the delay term; the reduction in storage is considerable when compared with conventional methods, such as fully or partially implicit methods. However, if the problem is extremely nonlinear it may happen that a lower-order method is more stable (see the example in Section 4.4).

(v) For several test examples we compared the GPC methods with an adjusted form of the ADI method. The storage reduction factors are roughly 5 and 2 for these problems (in favour of the GPC methods). Moreover, in terms of CP seconds (measured on a CDC 750 computer) the GPC methods are more efficient.

(vi) Finally, because of its explicit character, the GPC method can also be applied to non-5-point space discretizations which allows us to integrate problems with mixed derivatives, or to employ high-order space molecules; in the latter case, the magnitude of the spatial meshes can be increased resulting in a smaller spectral radius $B(\partial \mathbf{f}/\partial \mathbf{y})$ and as a consequence a smaller number of stages per step. This is in sharp contrast with the fully or partially implicit methods where these high-order molecules will considerably increase the computational effort to solve the implicit relations.

Acknowledgement

We are grateful to the referees for their constructive remarks.

References

ARTOLA, M. 1967 Equations paraboliques à retardement. *C.R. Acad. Sci. Paris* **264,** 668–671.

BAKER, C. T. H. 1977 The Numerical Treatment of Integral Equations. Oxford: Clarendon Press.

BARWELL, V. K. 1975 Special stability problems for functional differential equations. *BIT* **15,** 130–135.

BELLMAN, R., & COOKE, K. L. 1963 *Differential-Difference Equations*, New York: Academic Press.

CHOSKY, N. H. 1966 Time-lag controls: a bibliography. *IRE Transactions on Automatic Control* **AC-5,** 66–70.

CRYER, C. W. 1972 Numerical methods for functional differential equations. In *Delay and Functional Differential Equations and Their Applications* (K. Schmitt, Ed.), pp. 17–101. New York: Academic Press.

CRYER, C. W. 1974 Highly stable multistep methods for retarded differential equations. *SIAM J. Numer. Anal.* **11,** 788–797.

EL'SGOL'TS, L. E., & NORKIN, S. B. 1973 *Introduction to the Theory and Application of Differential Equations with Deviating Arguments.* [Translation by J. L. Casti], New York: Academic Press.

HOUWEN, P. J. VAN DER, & VERWER, J. G. 1979 One-step splitting methods for semi-discrete parabolic equations. *Computing* **22,** 291–309.

HOUWEN, P. J. VAN DER, & SOMMEIJER, B. P. 1983a Improved absolute stability of predictor-corrector methods for retarded differential equations. In: *Differential-difference Equations. ISNM* **62,** pp. 137–148. Basel: Birkhäuser-Verlag.

HOUWEN, P. J. VAN DER, & SOMMEIJER, B. P. 1983b Predictor-corrector methods with improved absolute stability regions. *IMA J. Numer. Anal.* **3,** 417–437.

LAMBERT, J. D. 1973 *Computational Methods in Ordinary Differential Equations.* London: John Wiley.

PEACEMAN, D. W. & RACHFORD JR., H. H. 1955 The numerical solution of parabolic and elliptic differential equations. *J. Soc. Ind. Appl. Math.*, **3,** 28–41.

STETTER, H. J. 1968 Improved absolute stability of predictor-corrector schemes. *Computing* **3,** 286–296.

TAVERNINI, L. 1973 Linear multistep methods for the numerical solution of Volterra functional differential equations. *J. Applicable Anal.* **3,** 169–185.

TRAVIS, C. C., & WEBB, G. F. 1974 Existence and stability for partial functional differential equations. *Transactions Amer. Math. Soc.* **200,** 395–418.

WANG, P. K. C. 1963 Asymptotic stability of a time-delayed diffusion system. *J. Appl. Mech. (Ser. E)* **30,** 500–504.

WANG, P. K. C. 1975 Optimal control of parabolic systems with boundary conditions involving time delays. *SIAM J. Control* **13,** 274–293.

WEISS, R. 1959 Transportation lag—an annotated bibliography. *IRE Transactions on Automatic Control* **AC-4,** 56–64.

WIEDERHOLT, L. F. 1976 Stability of multistep methods for delay differential equations. *Math. of Comp.* **30,** 283–290.