Preconditioning in implicit initial-value problem methods on parallel computers*

P.J. van der Houwen
CWI, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Received 31 October 1992

Abstract

Implicit step-by-step methods for numerically solving the initial-value problem $\{y' = f(y),\ y(0) = y_0\}$ usually lead to implicit relations of which the Jacobian can be approximated by a matrix of the special form $K = I - hM \otimes J$, where M is a matrix characterizing the step-by-step method and J is the Jacobian of f. Similar implicit relations are encountered in discretizing initial-value problems for other types of functional equations such as VIEs, VIDEs and DDEs. Application of (modified) Newton iteration for solving these implicit relations requires the LU-decomposition of K. If s and d are the dimensions of M and J, respectively, then this LU-decomposition is an $O(s^3d^3)$ process, which is extremely costly for large values of sd. We shall discuss parallel iteration methods for solving the implicit relations that exploit the special form of Jacobian matrix K. Their main characteristic is that each processor is required to compute LU-decompositions of matrices of dimension d, so that this part of the computational work is reduced by a factor s^3. On the other hand, the number of iterations in these parallel iteration methods is usually much larger than in Newton iteration. In this contribution, we will try to reduce the number of iterations by improving the convergence of such parallel iteration methods by means of preconditioning.

Keywords: Numerical analysis, implicit step-by-step methods, preconditioning, parallelism.

Subject classification: 65M10, 65M20.

1. Introduction

In this paper, we study parallel step-by-step methods for solving initial-value problems (IVPs) for a variety of functional equations, such as ordinary differential equations (ODEs), Volterra integral equations (VIEs), Volterra integro-differential equations (VIDEs), delay-differential equations (DDEs), etc. Our approach is the parallel iteration of the implicit relations associated with an *implicit* integration method (the corrector

*This paper is presented as an outcome of the LMS Durham Symposium convened by Professor C.T.H. Baker on 4–14 July 1992, with support from the SERC under grant reference number GR/H03964.

method). The corrector methods will be represented in the partitioned General Linear Method (GLM) form introduced by Burrage and Butcher [5]:

$$Y = F(t_n, Y_0, Y_1, \ldots, Y_n) + h^{\nu}(M \otimes I)G(t_n, Y), \quad Y_{n+1} = H(t_n, Y_0, Y_1, \ldots, Y_n, Y),$$
$$Y := (Y_1^{\mathrm{T}}, Y_2^{\mathrm{T}}, \ldots, Y_s^{\mathrm{T}})^{\mathrm{T}}, \quad Y_n := (Y_{n1}^{\mathrm{T}}, Y_{n2}^{\mathrm{T}}, \ldots, Y_{nr}^{\mathrm{T}})^{\mathrm{T}}, \quad n = 0, 1, 2, \ldots, \tag{1.1}$$

where ν is the order of the IVP, M is an s-by-s matrix with constant entries characterizing the corrector, Y and Y_n represent an s-dimensional and r-dimensional block vector of numerical approximations to the exact solution of the IVP. If the IVP has dimension d, then Y_j and Y are vectors in rd-dimensional and sd-dimensional vector spaces, respectively, and F, G and H are functions depending both on the IVP and the step-by-step method. Furthermore, $M \otimes I$ denotes the direct product of the matrices M and I.

In each step, the block vectors $\{Y_0, Y_1, \ldots, Y_n\}$ are the input vectors, Y_{n+1} is the internal stage vector or "blackbox" vector. We shall say that the method has s internal stages and r output points. The equations defining Y and Y_{n+1} are, respectively, called the stage vector equation and the output formula (or step point formula). Both equations have a "history" term containing only backvalues, and a "future" term containing the unknown stage vector Y.

Each step requires the solution of the stage vector equation in the corrector method (1.1). It will be assumed that the Jacobian of $(M \otimes I)G$ satisfies a relation of the form

$$(M \otimes I)\,\frac{\partial G(t_n, Y)}{\partial Y} = M \otimes J_n + o(h), \tag{1.2}$$

where J_n is a d-by-d matrix evaluated at a single point and completely determined by the IVP. Before discussing the iteration process for solving the stage vector equation, we first consider examples of step-by-step methods of the form (1.1). For simplicity of notation, all formulas refer to IVPs for *scalar* functional equations and we shall extensively use "componentwise" notation, that is, given a function $f: \mathbb{R} \to \mathbb{R}$, then $f(\boldsymbol{v})$ denotes the vector with entries $f(v_j)$.

1.1. ORDINARY DIFFERENTIAL EQUATIONS

Consider the IVP for ordinary differential equations (ODEs)

$$y^{(\nu)}(t) = f(t, y(t)), \quad y(t_0) = y_0, \quad t_0 \le t \le t_{\mathrm{end}}, \tag{1.3}$$

and consider the GLM as presented in [5]:

$$Y = C_{12}Y_n + h^{\nu}C_{11}f(t_n e + hc, Y),$$
$$Y_{n+1} = C_{22}Y_n + h^{\nu}C_{21}f(t_n e + hc, Y), \quad n = 0, 1, 2, \ldots . \tag{1.4}$$

Here, C_{12} is an s-by-r matrix, C_{11} is an s-by-s matrix, C_{22} is an r-by-r matrix, and C_{21} is an r-by-s matrix. This method can be cast into the form (1.1) with

$$F(Y_0,\dots,Y_n) = C_{12}Y_n, \quad MG(t_n,Y) = C_{11}f(t_ne + hc, Y),$$

$$H(Y_0,\dots,Y_n,Y) = C_{22}Y_n + h^{\nu}C_{21}f(t_ne + hc, Y),$$

so that (1.2) is satisfied with $M = C_{11}$ and $J_n = (\partial f/\partial y)_n$.

1.2. SECOND-KIND VOLTERRA EQUATIONS

Consider the second-kind Volterra equation (VIE)

$$y(t) = g(t) + \int_0^t K(t, x, y(x))\,\mathrm{d}x, \quad 0 \le t \le T, \tag{1.5}$$

and the Volterra Runge–Kutta (VRK) method (cf. [1])

$$\begin{aligned} Y &= F^*(t_ne + hc, t_ne) + h\Phi(Y), \\ \Phi_i(t_n, Y) &:= a_i^{\mathrm{T}}K(t_ne + hc_ie, t_ne + hc, Y), \qquad i = 1,\dots,s, \\ y_{n+1} &= F^*(t_n + h, t_n) + hb^{\mathrm{T}}K(t_ne + he, t_ne + hc, Y). \end{aligned} \tag{1.6}$$

The VRK parameters are stored in the s-dimensional vectors a_i, b and c, and e denotes the unit vector $(1,\dots,1)^{\mathrm{T}}$. $F^*(t, s)$ denotes a numerical approximation to the lag term

$$F(t,s) := g(t) + \int_0^s K(t, x, y(x))\,\mathrm{d}x \tag{1.7}$$

using only the imput values $\{y_0, y_1, \dots, y_n\}$. The method (1.6) fits into the class (1.1) with $r = 1$, $Y_n = y_n$, and

$$F(t_n, Y_0,\dots,Y_n) := F^*(t_ne + h\theta, t_ne), \quad MG(t_n, Y) := \Phi(t_n, Y),$$

$$H(t_n, Y_0,\dots,Y_n,Y) := F^*(t_n + h, t_n) + hb^{\mathrm{T}}K(t_ne + he, t_ne + hc, Y).$$

Let A be the matrix with row vectors a_i^{T}. Then, $M = A$ and $J_n = (\partial K/\partial y)_n$.

Another class of VIE methods that can be presented in the form (1.1) is the block version of Volterra Linear Multistep method for the VIE (1.5)

$$\begin{aligned} Y_{n+1} &= AY_n + BF^*(t_ne + h\theta, t_ne + ha) \\ &\quad + hCK(t_ne + hb, t_ne + hc, Y_{n+1}), \quad n = 0,\ 1,\ 2,\dots, \end{aligned} \tag{1.8}$$

where θ, a, b and c are k-dimensional vectors, A, B and C are k-by-k matrices, and where $F^*(t, s)$ denotes a numerical approximation to the lag term (1.7) using only values of the input vectors $\{Y_0, Y_1, \dots, Y_n\}$. By setting $r = s$ and defining

$$F(t_n, Y_0, \dots, Y_n) := AY_n + BF^*(t_n e + h\theta, t_n e + ha),$$
$$MG(t_n, Y) := CK(t_n e + hb, t_n e + hc, Y),$$
$$H(t_n, Y_0, \dots, Y_n, Y) := Y,$$

the method (1.8) fits into the class (1.1) and (1.2) is satisfied with $M = C$ and $J_n = (\partial K/\partial y)_n$.

1.3. VOTERRA INTEGRO-DIFFERENTIAL EQUATIONS

Consider the IVP for Volterra integro-differential equations

$$\begin{aligned} y'(t) &= f(y(t)) + z(t), \\ z(t) &:= \int_0^t K(t, s, y(s))\, \mathrm{d}s, \\ y(0) &= y_0, \quad 0 \le y \le T, \end{aligned} \tag{1.9}$$

and the VDRK method (cf. [1])

$$\begin{aligned} Y &= y_n e + hAf(Y) + hA[Z(t_n e + hc) + h\Phi(t_n, Y)], \\ \Phi(t_n, Y) &:= (e_i^{\mathrm{T}} A^* K(t_n e + hd_i, t_n e + hc, Y)), \quad i = 1, \dots, s, \\ y_{n+1} &= y_n + b^{\mathrm{T}} A^{-1}(Y - y_n e), \end{aligned} \tag{1.10}$$

where C is assumed to be nonsingular and $Z(t)$ denotes a numerical approximation to the lag term

$$Z(t) := \int_0^t K(T, x, y(x))\, \mathrm{d}x \tag{1.11}$$

using only the input values $\{y_0, y_1, \dots, y_n\}$. The method (1.6) fits into the class (1.1) with $r = 1$, $Y_n = y_n$, and

$$F(t_n, Y_0, \ldots, Y_n) := y_n e + hAZ(t_n e + hc),$$

$$MG(t_n, Y) := A[f(Y) + h\Phi(Y))],$$

$$H(t_n, Y_0, \ldots, Y_n, Y) := y_n + b^{\mathrm{T}} A^{-1}(Y - y_n e).$$

Condition (1.2) is satisfied with $M = A$ and $J_n = (\partial f/\partial y)_n$.

1.4. DELAY-DIFFERENTIAL EQUATIONS

Consider the IVP for delay-differential equations

$$y'(t) = f(y(t)) + z(t), \quad z(t) := g(y(t - a(t)), \quad 0 \le t \le T, \tag{1.12}$$

where $a(t)$ is positive and $y(t)$ is prescribed for $t \le 0$. This IVP can be solved by the RK method

$$\begin{aligned} Y &= y_n e + hAZ(t_n e + he) + hAf(Y), \\ y_{n+1} &= y_n + b^{\mathrm{T}} A^{-1}(Y - y_n e), \end{aligned} \qquad n = 1, 2, \ldots, \tag{1.13}$$

where A is assumed to be nonsingular and $Z(t)$ denotes a numerical approximation to the delay term $z(t)$ using only the input values $\{y_0, y_1, \ldots, y_n\}$. The method (1.6) fits into the class (1.1) with $r = 1$, $Y_n = y_n$, and

$$F(t_n, Y_0, \ldots, Y_n) := y_n e + hCZ(t_n e + hc),$$

$$MG(Y) := Af(Y),$$

$$H(t_n, Y_0, \ldots, Y_n, Y) := y_n + b^{\mathrm{T}} A^{-1}(Y - y_n e).$$

Condition (1.2) is satisfied with $M = A$ and $J_n = (\partial f/\partial y)_n$.

2. Preconditioned parallel iteration methods

We shall study iterative methods for solving the stage vector equation on parallel computers. Let us write the stage vector equation in (1.1) in the form

$$R(t_n, Y) := Y - F(t_n Y_0, \ldots, Y_n) - h^{\nu}(M \otimes I)G(t_n, Y) = 0. \tag{2.1a}$$

A general format for describing well-known iteration methods for solving this nonlinear system is of the form

$$K_{nj}(Y^{(j)} = K_{nj}(Y^{(j-1)}) - R(t_n, Y^{(j-1)}, \quad j = 1, \ldots, m, \tag{2.1b}$$

where $Y^{(0)}$ is a given initial iterate and K_{nj} is an operator that may depend on the step point index n and the iteration index j. Evidently, if (2.1b) converges, then it converges to the stage vector Y. The most simple choice is $K_{nj} = I$ to obtain fixed-point iteration (or functional iteration). At the other end of the scale, we have the Newton iteration process with $K_{nj} = I - h^{\nu}(M \otimes I)\partial G(t_n, Y^{(j-1)})/\partial Y$. All kinds of "intermediate" choices are possible. In the case where the residual function $R_n(Y)$ originates from an implicit method for ODEIVPs, Burrage [3] discussed a number of such "intermediate" choices, among which a few cases that are suitable on parallel computers.

In this paper, we also exploit parallel computers to improve the rate of convergence of the iteration process (2.1b). However, our considerations will not be restricted to ODEIVPs, but we also consider general residual functions $R(t_n, Y)$ restricted only by the condition that the Jacobian of $G(t_n, Y)$ satisfies (1.2).

The most obvious way for obtaining fast convergence is the use of high-order predictors. In fact, Burrage gave a detailed analysis of the effect of the predictors on the overall accuracy and showed that it is advisable to use higher-order predictor formulas for computing the initial iterate $Y^{(0)}$ (see also [2] and [4]). One possibility for constructing a high-order, parallel predictor formula consists of applying the step-by-step method not only at step points, but also (in parallel) at off-step points, so that, in each step, a whole block of approximations to the exact solution is computed. These approximations can then be used in the next step for obtaining a high-order predictor formula. By choosing the off-step points narrowly spaced, we achieve a much more accurate predicted value than can be obtained by predictor formulas based on preceding step point values. For ODEs, the performance of such a block predictor approach was studied in [10] and shown to lead to substantial reduction of the number of iterations. It seems that this approach can also be applied to the more general step-by-step method (1.1).

Having determined a suitable predictor, we should try to exploit parallel architectures for improving the convergence of the iteration process. This will be the subject of the rest of this paper.

We shall consider operators K_{nj} of the form

$$K_{nj}(Y^{(j)}) := P_j^{-1}[I - h^{\nu}(T \otimes J_n)]. \tag{2.1c}$$

Here, P_j is a real sd-by-sd matrix, T is a real s-by-s matrix, and J_n is the d-by-d matrix determined by (1.2) and assumed to be evaluated at (t_n, Y_n, e). Substitution of (2.1c) into (2.1b) yields

$$[I - h^{\nu}(T \otimes J_n)]Y^{(j)} = [I - h^{\nu}(T \otimes J_n)]Y^{(j-1)} - P_j R(t_n, Y^{(j-1)}), \quad j = 1, \ldots, m.$$

Let us first consider the case $P_j = I$. For $T = O$ and $T = M$, we easily recognize standard fixed-point iteration and standard modified Newton iteration, respectively. For non-stiff problems, fixed-point iteration is an efficient iteration method, particularly on parallel computers because of its intrinsic parallelism (cf. [17,13,11]). In the case of stiff problems, fixed-point iteration cannot be used. On the other hand, setting $T = M$, the computational complexity is quite large and cannot be sufficiently reduced on parallel computers. Therefore, instead of $T = M$, we choose for T a triangular s-by-s matrix of the form

$$T = \begin{pmatrix} D_1 & O & O & \cdots \\ F_1 & D_2 & O & \cdots \\ F_2 & F_3 & D_3 & \cdots \\ & \cdots & \cdots & \end{pmatrix}, \tag{2.1d}$$

where the D_i are diagonal s^*-by-s^* matrices and the F_i are allowed to be full s^*-by-s^* matrices. Recalling thay Y is a vector with s vector components of the IVP dimension d, we see that matrices T of the form (2.1d) demand s^* processors to compute the first s^* vector components of Y in parallel, next these s^* processors compute the second set of s^* vector components of Y, etc. Thus, each iteration requires the solution of s/s^* implicit relations of dimension d per processor. The resulting iteration process will be called *diagonally implicit iteration.* We remark that for nonstiff problems, we may set $T = O$, so that $s^* = s$ processors and s parallel evaluations of the s vector components of $R_n(Y^{(j-1)})$ per iteration are required. As far as we know, until now only the case $s^* = s$ (i.e. T diagonal) has been considered [12]. It would be of interest to investigate whether more efficient choices of T using nonzero F_i are possible. Another possibility is to rearrange the components of the stage vector Y (and the stage vector iterates $Y^{(j)}$) in d blocks of s components where the ith block contains the ith component of the s vectors Y_j, $j = 1, \ldots, s$. Using diagonal matrices T would again lead to a diagonally implicit iteration process, now using d processors where each iteration requires the solution of one implicit relation of dimension s per processor. Iteration methods of this type will be analysed in a forthcoming paper.

If $P_j \neq I$, then P_j may be considered as a preconditioner for the residual function $R_n(Y)$. In time marching techniques for solving steady boundary value problems, the use of preconditioners for the conditioning of residual terms has frequently been used in the literature [15,14,17,18,9]. The possibility of preconditioning residual terms in diagonally implicit iteration methods was suggested already in [3] and [8], and will be elaborated in this paper.

In order to see the effect of preconditioning on the iteration error $Y^{(j)} - Y$, we consider the error equation

$$[I - h^\nu(T \otimes J_n)]\ [Y^{(j)} - Y] = h^\nu P_j(M \otimes I)[G(t_n, Y^{(j-1)}) - G(t_n, Y)]$$
$$+ [I - P_j - h^\nu(T \otimes J_n)][Y^{(j-1)} - Y]. \tag{2.2}$$

We shall confine our considerations to the case where G is linear in Y, satisfying the relation

$$(M \otimes I)[G(t_n, U) - G(t_n, V)] = (M \otimes J)[U - V],$$

where J is a constant d-by-d matrix. On substitution into (2.2) and setting $J_n = J$, we obtain the linear recursion

$$\begin{aligned} Y^{(j)} - Y &= Z_j[Y^{(j-1)} - Y], \\ Z_j &:= [I - h^{\nu}(T \otimes J)]^{-1}(I - h^{\nu}(T \otimes J) - P_j[I - h^{\nu}(M \otimes J)]). \end{aligned} \tag{2.3}$$

We remark that in many nonlinear problems, this linear error equation presents an $O(h^{\nu+1)})$ approximation to the nonlinear error eq. (2.2).

We shall consider the recursion (2.3) in the cases where the matrix P_j is such that Z_j has one or more zero eigenvalues (spectral fitting) or Z_j behaves as $O(h^{2\nu})$.

2.1. SPECTRAL FITTING

Suppose that we choose

$$P_j = [I - h^{\nu}\omega_j(T \otimes I)][I - h^{\nu}\omega_j(M \otimes I)]^{-1}, \tag{2.4}$$

where ω_j is a fixed point independent of h on the nonpositive real axis. Such preconditioners require the inversion of m s-by-s matrices each time the value of $h^{\nu}\omega_j$ is changed. Since s is usually small ($s \le 5$), the computational costs are hardly increased.

The convergence of {(2.3), (2.4)} can be investigated by considering a scalar IVP $y^{(\nu)} = \lambda y$, where λ runs through the eigenvalues of the matrix J_n of the original problem. Setting $J = \lambda$ in eqs. {(2.3), (2.4)} yields

$$\begin{aligned} Y^{(j)} - Y &= Z_j(h, \lambda)[Y^{(j-1)} - Y], \\ Z_j(h, \lambda) &:= h^{\nu}(\omega_j - \lambda)(I - h^{\nu}\lambda T)^{-1}(T - P_j M). \end{aligned}$$

Evidently, we have convergence if h is sufficiently small. In particular, we have strong damping in the neighbourhood of the point ω_j. This iteration method will be said to be *fitted at the point* ω_j. We remark that the preconditioner (2.4) allows only *real* fitting points. By replacing the iteration scheme (2.1) by a three-term recursion, it is possible to allow complex fitting points.

If ω_j vanishes (i.e $P_j = I$), then $Z_j(h, \lambda) := -h^{\nu}\lambda(I - h^{\nu}\lambda T)^{-1}(T - M)$. This leads to amplification factors

$$\alpha(h^{\nu}\lambda) := h^{\nu}|\lambda|\rho((I - h^{\nu}\lambda T)^{-1}(M - T)), \tag{2.5}$$

where $\rho(A)$ indicates the spectral radius of a matrix A. It is our aim to reduce the magnitude of the amplification factor $\alpha(h^\nu\lambda)$ by a judicious choice of ω_j.

In the following two subsections, we consider the effect of spectral fitting in a single iteration. In subsection 2.1.3, the effect of a sequence of different values ω_j is investigated. In both cases, we set $T = \tau I$, with τ a given, nonnegative number.

2.1.1. *The effect of a single iteration for stiff problems*

In the case of stiff problems, we should use nonzero values for τ. A suitable choice of τ minimizes the spectral radius of $Z_j(h, \infty) = -[I - \tau^{-1}M][I - h^\nu\omega_j M]^{-1}$ (cf. [11]). Assuming that τ is defined, we want to determine the region where the use of preconditioning results in stronger damping than obtained by applying unpreconditioned iteration. This region will be called the *region of improved damping*.

THEOREM 2.1

Let M have its eigenvalues μ in the positive halfplane. Then, the region of improved damping consists of the exterior of the union of disks $D_0(\mu, \omega_j)$ defined by

$$D_0(\mu, \omega_j) := \left\{\lambda;\left|\lambda + \frac{\omega_j}{c_j}\right| \le \frac{\omega_j}{c_j}|1 - h^\nu\omega_j\mu|\right\},$$

$$c_j := |1 - h^\nu\omega_j\mu|^2 - 1.$$

Proof

The eigenvalues of the iteration matrix $Z_j(h, \lambda)$ corresponding to μ are given by

$$\zeta(\mu, \omega_j, \lambda) := h^\nu(\lambda - \omega_j)(\tau - \mu)(1 - h^\nu\lambda\tau)^{-1}(1 - h^\nu\omega_j\mu)^{-1}.$$

The region where the use of preconditioning results in stronger damping is determined by the inequality $|\zeta(\mu, \omega_j, \lambda)| < |\zeta_j(\mu, 0, \lambda)|$, i.e. by $|\lambda - \omega_j| < |\lambda||1 - h^\nu\omega_j\mu|$. Recalling that ω_j is assumed to be nonpositive and using $\text{Re}(\mu) \ge 0$, it can straightforwardly be shown that this inequality is satisfied outside the disks $D_0(\mu, \omega_j)$. □

Thus, if the spectrum of J_n is in the left halfplane and if ω_j is chosen to be negative, then this theorem indicates that preconditioning should result in better damping of those error components that do not correspond to eigenvalues close to the imaginary axis.

Next, we consider the region where the amplification factors are bounded by a prescribed number α. To this end, we introduce the quantities

$$\delta := \frac{\alpha}{|\tau - \mu| h^\nu} \sqrt{1 - 2h^\nu \omega_j \mathrm{Re}(\mu) + h^{2\nu}\omega_j^2 |\mu|^2},$$

$$\lambda_j := \frac{\omega_j - \tau\delta^2}{1 - \tau^2\delta^2 h^{2\nu}}, \tag{2.6a}$$

$$\delta_j(\alpha, \mu, \omega_j) := \sqrt{\lambda_j^2 + \frac{\delta^2 - \omega_j^2}{1 - \tau^2\delta^2 h^{2\nu}}},$$

and we define the disk $D(\alpha, \mu, \omega_j)$ of radius δ_j and centered at λ_j:

$$D(\alpha, \mu, \omega_j) := \{\lambda : |\lambda - \lambda_j| \leq \delta_j(\alpha, \mu, \omega_j)\}. \tag{2.6b}$$

The following theorem is now straightforwardly proved.

THEOREM 2.2

The eigenvalues $\zeta(\mu, \omega_j, \lambda)$ are bounded by α if either $\lambda \in D(\alpha, \mu, \omega_j)$ and $|\tau\delta h^\nu| \leq 1$ or if $l \notin D(\alpha, \mu, \omega_j)$ and $|\tau\delta h^\nu| \geq 1$. □

From this theorem, we conclude that for $\omega_j \leq 0$ and for small values of α (i.e. small values of δ), the amplification factors are bounded by α *inside* the disks $D(\alpha, \mu, \omega_j)$ centered on the negative axis. The radius of these disks increases as α increases. If δ becomes larger than $(\tau h^\nu)^{-1}$ (assuming that $\tau \neq 0$), then the disks $D(\alpha, \mu, \omega_j)$ are centered on the positive axis and the amplification factors are bounded by α *outside* these disks.

In the case of *stiff* problems, it is of particular interest to consider the case where the stepsize is relatively large, so that we have to take into account the amplification factors in the whole left halfplane. A maximal amplification factor α in the whole left-hand side plane can be obtained by requiring that $\lambda_j \geq \delta_j$ and $|\tau\delta h^\nu| \geq 1$. On substitution of (2.6a), we obtain the condition

$$\alpha^2[1 - 2h^\nu\omega_j \mathrm{Re}(\mu) + h^{2\nu}\omega_j^2|\mu|^2] \geq |\tau - \mu|^2 \max\left\{h^{2\nu}\omega_j^2, \frac{1}{\tau^2}\right\}.$$

This leads us to the following result:

THEOREM 2.3

The maximal amplification factor in the left-hand side plane is given by

$$\alpha(h^\nu\omega_j) := \max_\mu \frac{|1 - \mu\tau^{-1}| \max(h^\nu\tau|\omega_j|, 1)}{\sqrt{1 - 2h^\nu\omega_j \mathrm{Re}(\mu) + h^{2\nu}\omega_j^2|\mu|^2}}. \qquad □$$

EXAMPLE 2.1

Consider the two-point Gauss–Legendre corrector for first-order ODEs ($\nu = 1$), where

$$M = \frac{1}{12}\begin{pmatrix} 3 & 3 - 2\sqrt{3} \\ 3 + 2\sqrt{3} & 3 \end{pmatrix}.$$

Hence, $\mathrm{Re}(\mu) = 1/4$ and $|\mu|^2 = 1/12$. Without preconditioning, we find $\alpha(h\omega_j) = |1 - \mu\tau^{-1}|$. By applying preconditioning, we find

$$\alpha(h\omega_j) = \frac{|1 - \mu\tau^{-1}|\max(h\tau|\omega_j|, 1)}{\sqrt{1 - h\omega_j/2 + h^2\omega_j^2/12}} = \frac{\max(h\tau|\omega_j|, 1)}{\sqrt{1 - h\omega_j/2 + h^2\omega_j^2/12}}\,\alpha(0).$$

For a typical value of $\tau = 1/4$ (cf. [12]), the reduction factor varies from 1 for $h\omega_j = 0$ to 0.48 for $h\omega_j = -4$. □

2.1.2. *The effect of a single iteration for nonstiff problems*

For nonstiff problems, we may use $T = O$ to obtain fixed-point iteration. Since theorems 2.1 and 2.2 also apply to the case $T = O$ by setting $\tau = 0$, we can achieve strong damping in prescribed disks $D(\alpha, \mu, \omega_j)$ by a suitable choice of ω_j and h. The analogue of theorem 3.3 becomes:

THEOREM 2.4

If $\tau = 0$ and $\omega_j = -\theta\rho$, $\theta \geq 0$, ρ denoting the spectral radius of J_n, then the maximal amplification factor in the disk $D(\alpha, \mu, \omega_j)$ is bounded by

$$\alpha(\mu, \theta, h^\nu\rho) := \frac{\sigma(\theta)|\mu|h^\nu\rho}{\sqrt{1 + 2\theta\,\mathrm{Re}(\mu)h^\nu\rho + \theta^2|\mu|^2h^{2\nu}\rho^2}},$$

$$\sigma(\theta) := \frac{\max_\lambda|\lambda - \omega_j|}{\rho}.$$ □

The spectrum function $\sigma(\theta)$ depends on the type of spectrum of the Jacobian matrix and determines the magnitude of $\alpha(\mu, \theta, h^\nu\rho)$. We remark that in the unpreconditioned case ($\theta = 0$), we have $\gamma(0) = 1$, so that the amplification factors are given by $|\mu|h^\nu\rho$, that is, they do not depend on the type of spectrum. By applying preconditioning, the particular type of spectrum can be exploited. For future reference, table 1 presents the functions $\sigma(\theta)$ for a few special cases.

Table 1

Spectrum function $\gamma(\theta)$, $\theta := -\omega_j/\rho$.

Half disk spectrum	$\{\lambda: \lvert\lambda\rvert \le \rho,\ \mathrm{Re}(\lambda) \le 0\}$	$\sigma(\theta) := \sqrt{1+\theta^2}$,	$\theta \ge 0$
Interval spectrum	$\{\lambda: -\rho \le \lambda \le 0\}$	$\sigma(\theta) := \max\{\theta, \lvert 1-\theta\rvert\}$,	$\theta \ge 0$
Full disk spectrum	$\{\lambda: \lvert\lambda - \theta_0\rho\rvert \le \rho,\ 1/2 \le \theta_0 \le 1\}$	$\sigma(\theta) := 1-\theta$,	$0 \le \theta \le \theta_0$
		$\sigma(\theta) := 1+\theta-2\theta_0$,	$\theta \ge \theta_0$

EXAMPLE 2.2

Consider again the two-point Gauss–Legendre corrector (compare example 2.1). The corresponding amplification factors are given by

$$\alpha(\mu, \theta, h\rho) := \frac{\sigma(\theta)h\rho}{\sqrt{12 + 6\theta h\rho + \theta^2 h^2 \rho^2}}.$$

Let us consider the case of a disk spectrum and let us choose $\theta = \theta_0$. Then,

$$\alpha(\mu, \theta_0, h\rho) := \frac{(1-\theta_0)h\rho}{\sqrt{12 + 6\theta_0 h\rho + \theta_0^2 h^2 \rho^2}}, \quad \tfrac{1}{2} \le \theta_0 \le 1.$$

Without preconditioning, we have $\alpha(\mu, 0, h\rho) = h\rho/\sqrt{12}$, so that the amplification factors of the preconditioned iteration method are smaller by at least a factor $1-\theta_0$. □

It follows from theorem 3.4 that $\alpha(\mu, \theta, h^\nu\rho) < \gamma(\theta)/\theta$ for any value of $h^\nu\rho$ and all values of μ in the right-hand side plane. Hence, if we can choose θ such that $\sigma(\theta) \le \theta$, then we have unconditional convergence for correctors with $\mathrm{Re}(\mu) \ge 0$ (for example, in the case of an interval spectrum with $\theta = 1/2$). In all other cases, we have to satisfy a convergence condition for the stepsize h. We shall derive an upper bound for h such that we have prescribed damping α of all error components corresponding to the eigenvalues λ of J_n. From theorem 2.4, it follows that h should satisfy the inequality

$$\alpha^2(1 + 2\theta\,\mathrm{Re}(\mu)\rho h^\nu + \theta^2 \lvert\mu\rvert^2 \rho^2 h^{2\nu}) \ge (\sigma(\theta)\lvert\mu\rvert\rho)^2 h^{2\nu}.$$

This leads to

$$(\sigma^2(\theta) - \alpha^2\theta^2)\lvert\mu\rvert^2\rho^2 h^{2\nu} - 2\alpha^2\mathrm{Re}(\mu)\rho h^\nu - \alpha^2 \le 0.$$

Assuming that $\sigma^2(\theta) - \alpha^2\theta^2$ is positive, this condition is fulfilled if h satisfies the convergence condition

$$\rho h^{\nu} \leq \frac{\alpha^2\theta\,\mathrm{Re}(\mu) + \sqrt{[\alpha\sigma(\theta)\,|\mu|]^2 - [\alpha^2\theta\,\mathrm{Im}(\mu)]^2}}{(\sigma^2(\theta) - \alpha^2\theta^2)\,|\mu|^2}.$$

Hence, we have the corollary:

COROLLARY 2.1

Let the conditions of theorem 2.4 be satisfied and let $\gamma^2(\theta) > \alpha^2\theta^2$. Then the spectral radius of the iteration matrix $Z(h, \lambda)$ is bounded by α for all h satisfying the inequality

$$h^{\nu} \leq \frac{\gamma(\alpha,\theta)}{\rho},\quad \gamma(\alpha,\theta) := \max_{\mu}\left\{\frac{\alpha^2\theta\,\mathrm{Re}(\mu) + \sqrt{[\alpha\sigma(\theta)\,|\mu|]^2 - [\alpha^2\theta\,\mathrm{Im}(\mu)]^2}}{(\sigma^2(\theta) - \alpha^2\theta^2)\,|\mu|^2}\right\}. \;\square$$

For small values of α, the convergence boundary $\gamma(\alpha, \theta)$ reduces to the simple expression

$$\gamma(\alpha,\theta) \approx \frac{\alpha}{\sigma(\theta)\rho(M)},\qquad \alpha \ll 1.$$

Thus, for small values of α, the convergence boundary is increased by a factor $1/\sigma(\theta)$ by applying preconditioning. In table 2, we present convergence boundaries for the case where M is defined by the stage vector matrix in the Butcher array of a few well-known Runge–Kutta methods for first-order ODEs.

Table 2

Convergence boundaries $\gamma(\alpha, \theta)$ for various values of (α, θ).

		Half disk spectrum				Interval spectrum			
Corrector	p	$(\alpha, 0)$	$(1, 1/2)$	$(1/2, 1/2)$	$(\alpha \ll 1, 1/2)$	$(1, 0)$	$(1/2, 1/2)$	$(1/4, 1/2)$	$(\alpha \ll 1, 1/2)$
Radau IIA	3	2.4α	3.6	1.3	2.1α	2.4α	4.4	1.5	4.8α
Gauss–Legendre	4	3.4α	5.2	1.9	3.0α	3.4α	6.4	2.2	6.8α

2.1.3. Chebyshev preconditioning

Let $\{\omega_j\}$ be a given set of fitting points on the negative axis and consider the *accumulated* amplification factor

$$\begin{aligned}\zeta_m(\mu,\lambda) &:= \prod_{j=1}^{m} \zeta(\mu,\omega_j,\lambda)\\ &= [h^{\nu}(\tau-\mu)(1-h^{\nu}\lambda\tau)^{-1}]^m \prod_{j=1}^{m}(\lambda-\omega_j)(1-h^{\nu}\omega_j\mu)^{-1}.\end{aligned} \tag{2.7}$$

Ideally, we should choose the parameter vector $\{\omega_j\}$ such that $\max_\mu |\zeta_m(\mu, \lambda)|$ is minimized in the region of relevant values of λ (relevant values of λ are understood to be values corresponding to dominant error components). In general, exact solution of this minimax problem is not possible. However, by observing that for $\tau \geq 0$ the function $\max_\mu |\zeta_m(\mu, \lambda)|$ has no poles in the left-hand side plane and always has m zeros ω_j, we can approximately solve the minimax problem by identifying the set $\{\omega_j\}$ with the set of zeros of the minimax polynomial of degree m corresponding to the region of relevant values of λ. In the case where the relevant values of λ are in a real interval $[a, b]$, the approximate minimax solution is obtained by identifying $\{\omega_j\}$ with the set of zeros of the Chebyshev polynomial of the first kind shifted to $[a, b]$, i.e. the zeros ω_j of $\max_\mu |\zeta_m(\mu, \lambda)|$ should coincide with the zeros of the polynomial

$$S_m(\lambda) := T_m\left(\frac{2\lambda - a - b}{a - b}\right).$$

For other regions of relevant values of λ, approximations to the minimax solution can be obtained by employing Faber polynomials. Coefficients of the approximating polynomials for circular sectors, including the important case of a half disk region, were given by Coleman and Smith [6]. However, the zeros of these polynomials need not be real, so that the iteration process should be modified (by using three-term recursions) in order to achieve fitting at complex points. This will be the subject of future research. In this paper, we confine our considerations to minimization on the interval $[a, b]$. This leads us to the *Chebyshev preconditioner* defined by

$$\begin{aligned} P_j &= [I - h^\nu \omega_j (T \otimes I)](I - h^\nu \omega_j (M \otimes I))^{-1}, \\ \omega_j &= \tfrac{1}{2}\left[-(a - b)\cos\left(\frac{(2j - 1)\pi}{2m}\right) + a + b\right], \end{aligned} \qquad j = 1, \ldots, m. \quad (2.8)$$

The following theorem presents an upper bound for the corresponding *maximal averaged amplification factor* defined by

$$\alpha_m(\lambda) := \max_\mu \sqrt[m]{|\zeta_m(\mu, \lambda)|}.$$

THEOREM 2.5

If $\tau \geq 0$ and $b \leq 0$, then the following assertions hold for the Chebyshev preconditioner (2.8):

(a) In the interval $a \leq \lambda \leq b$, the maximal averaged amplification factor satisfies the inequality

$$\alpha_m(\lambda) \le \frac{\rho((1-h^\nu\lambda\tau)^{-1}(M-\tau I))}{\min\limits_{\mu}\left(|\mu| \sqrt[m]{\left|T_m\left(\dfrac{2(h^\nu\mu)^{-1}-a-b}{a-b}\right)\right|}\right)}.$$

(b) Let the eigenvalues of M be in the right-hand side plane, and let $-h^\nu(a+b)\rho(M)$ be sufficiently small. Then, in the interval $a \le \lambda \le b$, the maximal averaged amplification factor is bounded by α for all h satisfying the inequality

$$h^\nu \le \frac{\gamma(\alpha)}{b-a}, \quad \gamma(\alpha) \approx \frac{4\alpha}{\sqrt[m]{2}\rho(\tau I - M)}.$$

(c) Let the conditions of assertion (b) be satisfied. Then, Chebyshev preconditioning reduces the maximal averaged amplification factors by a factor $2^{1/m}(1-b/a)/4$. □

Proof

(a) Since $\tau \ge 0$ and $b \le 0$, we derive from (2.7) and the relation

$$\prod_{j=1}^{m}(\lambda-\omega_j) = \text{const}\, T_m\left(\frac{2\lambda-a-b}{a-b}\right),$$

the following estimate for the averaged amplification factor:

$$\sqrt[m]{|\zeta_m(\mu,\lambda)|} = h^\nu|\tau-\mu||(1-h^\nu\lambda\tau)^{-1}|\sqrt[m]{\prod_{j=1}^{m}(\lambda-\omega_j)(1-h^\nu\omega_j\mu)^{-1}}$$

$$= |(1-h^\nu\lambda\tau)^{-1}||\tau\mu^{-1}-1|\sqrt[m]{\left|T_m\left(\frac{2\lambda-a-b}{a-b}\right)\right|\left|T_m\left(\frac{2(h^\nu\mu)^{-1}-a-b}{a-b}\right)\right|^{-1}}$$

In the interval $a \le \lambda \le b$, this estimate is bounded by $\alpha_m(a, b)$ as defined in the theorem.

(b) If $\mathrm{Re}(\mu) \ge 0$ and if $h^\nu(a+b)\rho(M) \ll 1$, then it is easily shown that

$$\alpha_m(\lambda) \le \tfrac{1}{4}\sqrt[m]{2}h^\nu(b-a)\rho((1-h^\nu\lambda\tau)^{-1}(M-\tau I))$$

$$\le \tfrac{1}{4}\sqrt[m]{2}h^\nu(b-a)\rho(M-\tau I).$$

Hence, to achieve a damping by a factor α in the interval $a \le \lambda \le b$, the stepsize should satisfy the condition given in part (b) of the theorem.

(c) From (2.5), we deduce that for unpreconditioned iteration and $a \le \lambda \le b$, the amplification factors satisfy the inequality

$$\alpha(h^{\nu}\lambda) = h^{\nu}|\lambda|\rho((1 - h^{\nu}\lambda\tau)^{-1}(M - \tau I)) \le h^{\nu}|a|\rho(M - \tau I). \qquad (2.5')$$

A comparison with the upper bound for $\alpha_m(\lambda)$ yields part (c) of the theorem. □

2.2. ORDER $h^{2\nu}$ ITERATION

So far, the amplification matrix Z in (2.3) has been of $O(h^{\nu})$ as h tends to zero. Suppose that we choose P such that Z becomes $O(h^{2\nu})$ as h tends to zero. It is easily verified that this can be achieved if P satisfies

$$P = I + h^{\nu}(M - T) \otimes J + O(h^{\nu+1}). \qquad (2.9)$$

It is possible to derive the condition for P ensuring that Z has an arbitrarily high order in h. However, this would be of value only in the case of problems where the Jacobian matrix is constant. In more general problems, there will be no advantage in using iteration methods of higher order than order two, so we will restrict the analysis to order $h^{2\nu}$ iteration.

Let us define

$$P = [I - h^{\nu}(T \otimes J_n)]^{-1}[I + h^{\nu}((M - 2T) \otimes J_n)], \qquad (2.10)$$

to obtain for the test equation $y^{(\nu)} = \lambda y$ the recursion

$$Y^{(j)} - Y = Z(h^{\nu}\lambda)[Y^{(j-1)} - Y],$$

$$Z(h^{\nu}\lambda) := h^{2\nu}\lambda^2[I - h^{\nu}\lambda T]^{-2}[M^2 - 2TM + T^2].$$

We shall refer to (2.10) as the $O(h^{2\nu})$ preconditioner. The corresponding amplification factors are given by

$$\alpha(h^{\nu}\lambda) := h^{2\nu}\lambda^2\rho((I - h^{\nu}\lambda T)^{-2}(M^2 - 2TM + T^2)), \quad \lambda \in \Lambda. \ (2.11)$$

In the case of nonstiff problems, we may set $T = O$, so that the amplification factors per iteration are given by $\alpha(h^{\nu}\lambda) = [h^{\nu}|\lambda|\rho(M)]^2$. A comparison with the amplification factors of iteration without preconditioning (see (2.5) with $T = O$) shows that we achieve in one iteration the same amplification factor as the unpreconditioned iteration method in two iterations. The price we have to pay for this second-order effect is twofold: the evaluation of the Jacobian matrix J during the integration process and the matrix–vector multiplication $(M \otimes J)R_n(Y^{(j-1)})$ in each iteration. However, from the definition of the residual term (2.1a), it follows that this price is quite reasonable, provided that

(i) the evaluation of J is "cheap" with respect to the evaluation of $F(t_n, Y_0, \ldots, Y_n)$,

(ii) the matrix–vector multiplication $(M \otimes J)R_n(Y^{(j-1)})$ is "cheap" with respect to the evaluation of $G(t_n, Y^{(j-1)})$.

Condition (i) is usually satisfied in the case of Volterra integral equations where $F(t_n, Y_0, \ldots, Y_n)$ contains the expensive lag term (hence, the parallel methods described in [7] can be accelerated by preconditioners of the form (2.10)). Moreover, in many problems there is no need to update the Jacobian in each step. Condition (ii) implies that the multiplication by the d-by-d matrix J should be "cheap" compared with the evaluation of a d-dimensional component of $G(t_n, Y^{(j-1)})$. For IVPs of moderate dimension, this requirement is usually fulfilled.

For stiff problems, we choose T such that the spectral radius of Z is minimized for $\lambda = \infty$, i.e. T minimizes the spectral radius of the matrix $T^{-2}M^2 - 2T^{-1}M + I$ (cf. subsection 2.1.1). Unlike nonstiff iteration, the amplification factors for stiff iteration with $T \neq O$ are not exact squares of the amplification factors defined by (2.5) (this would only be true if T and M commute). However, the $O(h^{2\nu})$ behaviour of Z should at least lead to a considerable improvement of the rate of convergence for the nonstiff error components. Since the Jacobian J_n and the LU-decomposition of $I - h^\nu(T \otimes J_n)$ are already available, we conclude that the application of $O(h^{2\nu})$ preconditioners does not increase the computational effort much, provided that

(iii) the forward/backward substitution associated with L and U, and the matrix–vector multiplication $((M - 2T) \otimes J_n)R_n(Y^{(j-1)})$ is "cheap" with respect to the evaluation of $G(t_n, Y^{(j-1)})$.

3. Numerical experiments

In this section, we report a few first numerical results obtained by introducing preconditioning in the parallel, iterated RK methods (PIRK methods, cf. [11]) for nonstiff, first-order IVPs,

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0. \tag{3.1}$$

The PC pair will be based on the last step value predictor and on Gauss correctors with Butcher arrays $\{A, \boldsymbol{b}, \boldsymbol{c}\}$. The PIRK method is then given by

$$\begin{aligned} Y^{(1)} &= t_n e + hP_1(A \otimes I)f(t_n e + \eta ch, y_n e), \\ Y^{(j)} &= Y^{(j-1)} - P_j[Y^{(j-1)} - y_n e - h(A \otimes I)f(t_n e + ch, Y^{(j-1)})], \quad j = 2, \ldots, m, \\ y_{n+1} &= y_n + h(\boldsymbol{b}^{\mathrm{T}} \otimes I)f(t_n e + ch, Y^{(m)}), \end{aligned} \tag{3.2}$$

where P_j is given by either (2.8) or (2.10). The parameter η is introduced to tune the arguments in the right-hand side function $f(t_n e + \eta ch, y_n e)$ in the case of nonautonomous problems. The most natural choice is such that applying the method to (3.1) yields the same results as applying the method to the autonomous form of (3.1). This leads to $\eta = 0$, and, in general, this value turns out to be slightly better in the case of unpreconditioned iteration and Chebyshev preconditioners. However, in the case of $O(h^2)$ preconditioners, the choice of $\eta = 1$ seems to be more effective.

In all experiments, the (fixed) stepsize is chosen such that the total number of sequential right-hand sides equals a prescribed number N_{seq} specified in the tables of results. The absolute error obtained at the end point of the integration interval is presented in the form 10^{-d} (d may be interpreted as the number of correct decimal digits). In order to test the order of the method, we also list the effective order defined by $p^* := [d(h) - d(2h)]/0.3$, h being the smallest stepsize used. The theoretical order p is given by $p = m + 1$ and $p = 2m + 1$ for spectral preconditioners and $O(h^2)$ preconditioners, respectively.

We shall present results for three choices of the preconditioner:

(i) No preconditioning (spectral preconditioning (2.4) with $T = O$ and $\omega_j = 0$).

(ii) Chebyshev preconditioning (2.8) with $T = O$ and $[a, b]$ determined by the range of the diagonal entries of (an approximation to) the Jacobian matrix.

(iii) $O(h^2)$ preconditioning (2.10) with $T = O$ and with J_n (an approximation to) the Jacobian matrix.

We do not at all claim that the interval $[a, b]$ as defined above is optimal for Chebyshev preconditioning, but it is reasonably cheap and it reflects some information on the location of the eigenvalues (in fact, for strongly diagonally dominant Jacobian matrices, the above choice should be quite good). The determination of generally effective parameters a and b will be the subject of future research.

3.1. SCALAR TEST PROBLEM

We start with the scalar test problem

$$\begin{gathered} y'(t) = -[1 + \sin(t + y^5(t))][y - \exp(-t)] - \exp(-t), \\ y(0) = 1, \quad 0 \le t \le 5, \end{gathered} \tag{3.3}$$

with exact solution $y(t) = \exp(-t)$. Along the exact solution, the derivative of the right-hand side function is given by $\partial f/\partial y = -1 - \sin(t + y^5)$. Table 3 shows that the Chebyshev preconditioner yields a spectacular improvement of the accuracy when compared with unpreconditioned iteration. This is of course due to the fact that Chebyshev preconditioners are more effective as the eigenvalue interval is smaller. The $O(h^2)$ preconditioner yields a less spectacular improvement, but is still quite effective for larger stepsizes (* indicates divergence and the highest accuracy obtained for equal numbers of sequential function calls are given in boldface in table 3).

Table 3

Correct decimal digits at $t = 5$ for problem (3.3) using the eighth-order Gauss corrector.

Preconditioner	η	m	$N_{seq} = 8$	$N_{seq} = 16$	$N_{seq} = 32$	$N_{seq} = 64$	p^*
(2.4) with $\omega = 0$	0	7	*	0.9	3.9	6.6	9.0
	1	7	*	0.8	3.6	6.4	9.3
(2.8) with $a = b = (\partial f/\partial y)_n$	0	7	2.9	**4.7**	**8.3**		12.0
	1	7	**3.2**	**4.7**	7.7		8.3
(2.10) with $J_n = (\partial f/\partial y)_n$	0	3	*	2.5	4.5	6.6	7.0
	1	3	*	2.7	4.9	7.1	7.3

3.2. SPECTRAL FITTING AT POSITIVE POINTS

Our second test problem is again a scalar initial-value problem

$$\begin{aligned} y'(t) &= \sin(y^5) - \sin(\cos^5(t)) - \sin(t), \\ y(0) &= 0, \qquad 0 \le t \le 2, \end{aligned} \tag{3.4}$$

with exact solution $y(t) = \cos(t)$. The derivative is given by $\partial f/\partial y = 5y^4\cos(y^5)$ assuming only positive values. This example was chosen to see the effect of spectral fitting at positive points (we recall that the spectral fitting points ω_j were assumed to be nonpositive). Table 4 shows that the Chebyshev preconditioner is less robust, but if convergent, it again produces a considerable improvement, while the $O(h^2)$ preconditioner yields improved but erratic results.

Table 4

Correct decimal digits at $t = 2$ for problem (3.4) using the eighth-order Gauss corrector.

Preconditioner	η	m	$N_{seq} = 8$	$N_{seq} = 16$	$N_{seq} = 32$	$N_{seq} = 64$	p^*
(2.4) with $\omega = 0$	0	7	1.9	3.4	4.8	7.0	7.3
	1	7	2.0	3.7	5.6	7.5	6.3
(2.8) with $a = b = (\partial f/\partial y)_n$	0	7	*	**6.5**	**9.0**		8.3
	1	7	*	5.4	7.2		6.0
(2.10) with $J_n = (\partial f/\partial y)_n$	0	3	2.5	3.1	4.9	6.7	6.0
	1	3	**4.4**	4.4	6.5	8.3	6.0

3.3. AUTONOMOUS SYSTEM

Next, we consider the autonomous system of equations

$$y'(t) = A(y(t))y(t) - \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

$$A(y) := \begin{pmatrix} -1 & \cos(y_1) \\ -\cos(y_2) & -2 \end{pmatrix}, \tag{3.5}$$

$$y(0) = 0, \qquad 0 \le t \le 2,$$

where the exact solution values at $t = 2$ are approximately given by $y(2) = (-0.954439856927, -0.071572789676)^T$. Let us approximate the Jacobian of this system by the matrix A. It can be verified numerically that its eigenvalues are complex with constant real part equal to $-3/2$, so that the diagonal-entries-range $[-2, -1]$ is far from optimal. Nevertheless, table 5 shows that Chebyshev preconditioning is surprisingly effective. Furthermore, taking into account that the matrix A is a rather poor approximation to the true Jacobian matrix, we may conclude that the $O(h^2)$ preconditioner leads to considerably improved results.

Table 5

Correct decimal digits at $t = 2$ for problem (3.5) using the eighth-order Gauss corrector.

Preconditioner	η	m	$N_{seq} = 8$	$N_{seq} = 16$	$N_{seq} = 32$	$N_{seq} = 64$	p^*
(2.4) with $\omega = 0$	–	7	0.9	4.2	6.7	9.2	8.3
(2.8) on $[a, b] = [-2, -1]$	–	7	4.2	6.6	9.0		8.0
(2.10) with $J_n = A_n$	–	3	3.2	5.8	8.6		9.3

3.4. NONAUTONOMOUS SYSTEM

Finally, consider the nonautonomous system

$$y'(t) = A(t, y(t))y(t) + g(t),$$

$$A(t, y) := \begin{pmatrix} -(1+t) & \sin(y_1) \\ \sin(y_2) & -(2-t) \end{pmatrix}, \qquad 0 \le t \le 2, \tag{3.6}$$

where the initial conditions and the source function $g(t)$ are determined by the exact solution $y(t) = (\sin(t), \cos(t))^T$. Again approximating the Jacobian matrix by the matrix A, it can be shown that initially the eigenvalues of A are complex with real part $-3/2$ and become both real at a point t between 1 and 3/2. At $t = 2$, they are close to -3.1 and 0.1. Table 6 presents results similar to the preceding example. Notice that it pays to use better approximations to the Jacobian in the $O(h^2)$ preconditioner.

Table 6

Correct decimal digits at $t = 2$ for problem (3.6) using the eighth-order Gauss corrector.

Preconditioner	η	m	$N_{seq} = 8$	$N_{seq} = 16$	$N_{seq} = 32$	$N_{seq} = 64$	p^*
(2.4) with $\omega = 0$	0	7	1.0	2.7	5.0	7.5	8.3
	1	7	0.4	2.4	5.1	8.2	10.3
(2.8) on $[a, b] = [-1-t, -2+t]$	0	7	3.3	4.1	6.3		7.3
	1	7	2.7	4.5	6.8		7.7
(2.10) with $J_n = A_n$	0	3	1.4	3.5	4.8	6.0	4.0
	1	3	1.5	3.2	4.8	6.1	4.3
(2.10) with $J_n = (\partial f/\partial y)_n$	0	3	2.1	3.5	5.2	7.0	6.0
	1	3	2.3	4.8	6.7		6.3

Acknowledgement

The author is grateful to Dr. B.P. Sommeijer for performing most of the experiments and for pointing out the accuracy affecting effect of the parameter η occurring in the first iteration of the iteration process (3.2).

References

[1] H. Brunner and P.J. van der Houwen, *The Numerical Solution of Volterra Equations* (North-Holland, Amsterdam, 1986).

[2] K. Burrage, The error behaviour of a general class of predictor–corrector methods, Appl. Numer. Math. (1990)201–216.

[3] K. Burrage, The search for the Holy Grail, or predictor–corrector methods for solving ODEIVPs, Appl. Numer. Math. (1992), to appear.

[4] K. Burrage, Efficient block predictor-corrector methods with a small number of iterations, J. Comp. Appl. Math. (1993), to appear.

[5] K. Burrage and J.C. Butcher, Nonlinear stability of a general class of differential equations methods, BIT 20(1980)185–203.

[6] J.P. Coleman and R.A. Smith, The Faber polynomials for circular sectors, Math. Comp. 49(1987) 81–84; 231–241.

[7] M.R. Crisci, P.J. van der Houwen, E. Russo and A. Vecchio, Block methods for second-kind Volterra equations, in preparation.

[8] P.J. van der Houwen, Parallel step-by-step methods, Appl. Numer. Math. 11(1993)69–81.

[9] P.J. van der Houwen, C. Boon and F.W. Wubs, Analysis of smoothing matrices for the preconditioning of elliptic difference equations, Z. angew. Math. Mech. 68(1988)3–10.

[10] P.J. van der Houwen and Nguyen huu Cong, Parallel block predictor–corrector methods of Runge–Kutta type, Report NM-R9220, Centre for Mathematics and Computer Science, Amsterdam (1992).

[11] P.J. van der Houwen and B.P. Sommeijer, Parallel iteration of high-order Runge–Kutta methods with stepsize control, J. Comp. Appl. Math. 29(1990)111–127.

[12] P.J. van der Houwen and B.P. Sommeijer, Iterated Runge–Kutta methods on parallel computers, SIAM J. Sci. Stat. Comput. 12(1991)1000–1028.

[13] K.R. Jackson and S.P. Nørsett, The potential for parallelism in Runge–Kutta methods, Part I: RK formulas in standard form, Technical Report No. 239/90, Department of Computer Science, University of Toronto (1990).

[14] A. Jameson, The evolution of computational methods in aerodynamics, J. Appl. Mech. 50(1983) 1052–1076.

[15] A. Lerat, Une classe de schémas aux différence implicites pour les systèmes hyperboliques de lois de conservation, C.R. Acad. Sci. Paris 288 (Série A) (1979)1033–1036.

[16] S.P. Nørsett and H.H. Simonsen, Aspects of parallel Runge–Kutta methods, in: *Numerical Methods for Ordinary Differential Equations, Proc.*, L'Aquila 1987, ed. A. Bellen, C.W. Gear and E. Russo, LNM 1386 (Springer, Berlin, 1989).

[17] E. Turkel, Acceleration to a steady state for the Euler equations, in: *Numerical Methods for the Euler Equations of Fluid Dynamics* (SIAM, Philadelphia, 1985) pp. 218–311.

[18] F.W. Wubs, Stabilization of explicit methods for hyperbolic partial differential equations, Int. J. Numer. Meth. Fluids 6(1986)641–657.