SIAM J. SCI. COMPUT.
Vol. 18, No. 1, pp. 41–55, January 1997

003

TRIANGULARLY IMPLICIT ITERATION METHODS FOR ODE-IVP SOLVERS*

P. J. VAN DER HOUWEN† AND J. J. B. DE SWART†

Abstract. It often happens that iteration processes used for solving the implicit relations arising in ODE-IVP methods only start to converge rapidly after a certain number of iterations. Fast convergence right from the beginning is particularly important if we want to use so-called step-parallel iteration in which the iteration method is concurrently applied at a number of step points. In this paper, we construct highly parallel iteration methods that do converge fast from the first iteration on. Our starting point is the PDIRK method (parallel, diagonally implicit, iterated Runge–Kutta method), designed for solving implicit Runge–Kutta equations on parallel computers. The PDIRK method may be considered as a Newton-type iteration in which the Newton Jacobian is "simplified" to block-diagonal form. However, when applied in a step-parallel mode, it turns out that its relatively slow convergence, or even divergent behavior, reduces the effectiveness of the step-parallel scheme. By replacing the block-diagonal Newton Jacobian approximation in PDIRK by a block-triangular approximation, we do achieve convergence right from the beginning at a modest increase of the computational costs. Our convergence analysis of the block-triangular approach will be given for the wide class of general linear methods, but the derivation of iteration schemes is limited to Runge–Kutta-based methods. A number of experiments show that the new parallel, triangularly implicit, iterated Runge–Kutta method (PTIRK method) is a considerable improvement over the PDIRK method.

Key words. numerical analysis, convergence of iteration methods, Runge–Kutta methods, parallelism

AMS subject classifications. 65L05, 65L06

PII. S1064827595287456

1. Introduction.

Suppose that we integrate the IVP

$$y' = f(y), \quad y(t_0) = y_0, \quad y, f \in \mathbf{R}^d$$

by an implicit step-by-step method. For the class of general linear methods (cf. Butcher [7, p. 367]), this requires in each step the solution of a nonlinear system of the form

$$R(Y) = 0, \quad R(Y) := Y - h(A \otimes I)F(Y) - W, \tag{1.1}$$

where A denotes a nonsingular $s \times s$ matrix, W is an sd-dimensional vector containing information computed in preceding integration steps, I is the $d \times d$ identity matrix, h is the stepsize $t_n - t_{n-1}$, and $\otimes$ denotes the Kronecker product. The s components Y_i of the sd-dimensional solution vector Y represent s numerical approximations to the s exact solution vectors $y(t_{n-1} + c_i h)$; here, the c_i denote the abscissas. Furthermore, for any vector $V = (V_i)$, $F(V)$ contains the derivative values $(f(V_i))$. It is assumed that the c_i are distinct. In the following, we shall use the notation I for any identity matrix. However, its order will always be clear from the context.

The solution Y of (1.1) will be called the stage vector and s will be referred to as the number of stages. The most well-known examples of step-by-step methods

*Received by the editors June 7, 1995; accepted for publication (in revised form) March 22, 1996. The research reported in this paper was partly supported by the Technology Foundation (STW) in the Netherlands.

http://www.siam.org/journals/sisc/18-1/28745.html

†CWI, P. O. Box 94079, 1090 GB Amsterdam, the Netherlands (P.J.van.der.Houwen@.cwi.nl, Jacques.de.Swart@cwi.nl).

that lead to implicit relations of the form (1.1) are provided by the class of implicit Runge–Kutta (RK) methods. In that case, s equals the number of *implicit* stages of the RK method. (Note that for RK methods having *explicit* stages, s is less than the *total* number of stages of the RK method, e.g., this happens for Lobatto methods.)

We want to solve the system (1.1), to be referred to as the corrector, by a parallel iteration process. Our starting point is the PDIRK method (Parallel, Diagonally implicit Iterated Runge–Kutta method) developed in [15]. The PDIRK method may be considered as a Newton-type iteration in which the Newton Jacobian is "simplified" to block-diagonal form, so that we have parallelism across the stages. Earlier comparisons on a four-processor ALLIANT FX/4 of the codes LSODE and RADAU5 (considered as belonging to the best sequential codes for stiff problems) with the PDIRK-based code PSODE (Parallel Software for ODEs) of Sommeijer [21] showed that in general PSODE is the most efficient one. On the average, it produces the same accuracy in half the CPU time as required by LSODE and RADAU5. In [16] the PDIRK method was applied in a step-parallel setting, where the iteration procedure is concurrently applied at a number of step points; that is, iteration at the point t_{n+1} is already started without waiting until the iterates $Y^{(j)}$ at t_n have converged. (For details and analysis of various step-parallel methods we refer to, e.g., [1], [2], [4], [5], [8], [11], [16], [17], [19].) This approach requires that the predictor formula needed to start iteration at t_{n+1} is based on a sufficiently "safe" iterate $Y^{(j)}$. In order to have an efficient step-parallel iteration process, the value of j for which $Y^{(j)}$ is sufficiently "safe" should be small, that is, substantially smaller than the order of the method (1.1). Thus, in the step-parallel approach it is particularly important that we are near convergence right from the beginning. Although the PDIRK iteration method is quite efficient when iterating until convergence, it does have the drawback of a rather slow initial convergence, and hence it is less suitable for combination with a step-parallel approach.

The aim of the present paper is to improve the initial convergence of the PDIRK method by replacing the Newton Jacobian $I - A \otimes hJ$ with a matrix $I - B \otimes hJ$, where B is *block-triangular* instead of block-diagonal as it is in the PDIRK method. (Here, J is an approximation to the Jacobian of the right-hand side function f at t_{n-1}.) The intrinsic parallelism of these "triangularly" iterated methods is hardly less than in the PDIRK methods. The approach for determining the triangular matrix B is the same as for the PDIRK methods and is based on minimizing the spectral radius of the amplification matrix for the stiff iteration errors. However, instead of a numerical search as used for PDIRK [15], B can now be computed by means of the Crout decomposition of A. Although the asymptotic speed of convergence of PDIRK and the new method are often comparable, it happens that the departure from normality of the amplification matrix is considerably less than for the new methods. Hence, we may expect faster initial convergence.

We tested the triangular iteration strategy on a few nonlinear problems from the literature. These experiments do show a considerable improvement of the initial speed of convergence for the new methods. Furthermore, we applied both methods to a rather difficult problem from circuit analysis and estimated CPU times on a four-processor Cray C98/4256 showing that the CPU time increases by less than 10%. A comparison with the RADAU5 code reveals that for this problem the new method is at least twice as fast.

2. The iteration scheme. Our starting point for solving the corrector equation (1.1) is the simplified (or modified) Newton iteration scheme

$$(I - A \otimes hJ)\Delta Y^{(j+1)} = -R(Y^{(j)}), \tag{2.1}$$

$$Y^{(j+1)} = Y^{(j)} + \Delta Y^{(j+1)}, \quad j = 0, 1, \ldots,$$

where J is an approximation to the Jacobian of the right-hand side function f at t_{n-1}, and $Y^{(0)}$ is the initial iterate to be provided by some predictor formula. Each iteration with (2.1) requires the solution of an sd-dimensional linear system for the Newton correction $\Delta Y^{(j+1)}$. In actual computation, the costs for solving this system can be reduced by first performing a similarity transformation of the iterates (cf. Butcher [6]) $Y^{(j)} = (Q \otimes I)X^{(j)}$, where Q is a nonsingular matrix. Q should be such that the system

$$(2.2) \quad \begin{aligned} (I - Q^{-1}AQ \otimes hJ)\Delta X^{(j+1)} &= -(Q^{-1} \otimes I)R(Y^{(j)}), \\ Y^{(j+1)} &= Y^{(j)} + (Q \otimes I)\Delta X^{(j+1)}, \quad j = 0, 1, \ldots, \end{aligned}$$

is easier to solve than (2.1).

For example, if A has positive eigenvalues, then the Schur decomposition of A has the form $A = QTQ^{-1}$, where Q is orthogonal and T is lower triangular with the eigenvalues of A on its diagonal. Hence, the linear system (2.2) is "triangularly implicit" and consists of s subsystems of dimension d that can be solved sequentially. On sequential computers, this is most effective if A has a one-point spectrum, so that only one LU decomposition is required (see, e.g., Burrage [3]). On parallel computers, the condition on the spectrum of A can be relaxed to requiring that A is nondefective and has arbitrary positive eigenvalues. Since the s LU decompositions can be computed in parallel, only one decomposition per processor is required. Similarly, in each iteration, the s forward–backward substitutions for the diagonal blocks and the s components of $R(Y^{(j)})$ can also be computed in parallel. RK methods whose RK matrices have *positive* eigenvalues can be found in Orel [20].

Unfortunately, the most powerful implicit methods (with respect to order of accuracy and stability) have matrices A with *complex* eigenvalues. One option to deal with the complex eigenvalue case is to decompose A into a real block-triangular matrix of which the diagonal blocks are either diagonal or 2×2 matrices. This leads to an sd-dimensional system that can be split into a sequence of subsystems of either dimension d or dimension $2d$. (This approach was followed in the implementation of the RADAU5 code of Hairer and Wanner [14].) A block-diagonal structure of $Q^{-1}AQ$ implies that (2.2) is suitable for implementation on a parallel system.

Another option for reducing computational costs that will be the subject of this paper replaces the matrix A in (2.1) by a "more convenient" matrix B. In this paper, we consider the case where B is lower triangular, i.e., $B = L + D$, where L is strictly lower triangular and D is diagonal with positive diagonal entries d_{ii}. This leads to the iteration scheme

$$(2.3) \quad \begin{aligned} (I - D \otimes hJ)\Delta Y^{(j+1)} &= (L \otimes hJ)\Delta Y^{(j+1)} - R(Y^{(j)}), \\ Y^{(j+1)} &= Y^{(j)} + \Delta Y^{(j+1)}, \quad j = 0, 1, \ldots. \end{aligned}$$

In the case where L vanishes and (1.1) represents a Runge–Kutta (RK) method, the resulting iteration scheme is the PDIRK method mentioned in section 1. The method (2.3) requires LU decompositions of the $d \times d$ matrices $I - hd_{ii}J$, $i = 1, \ldots, s$, and, in each iteration, the evaluation of the residue $R(Y^{(j)})$, s forward–backward substitutions, and the matrix–vector multiplication $(L \otimes hJ)\Delta Y^{(j+1)}$. By expressing this multiplication in terms of F, the scheme (2.3) can be replaced by

$$(2.4) \quad (I - D \otimes hJ)\Delta Y^{(j+1)} = h(L \otimes I)(F(Y^{(j+1)}) - F(Y^{(j)})) - R(Y^{(j)}).$$

This version may yield better convergence if the right-hand side Jacobian is a less accurate approximation of the true Jacobian. Just like the scheme (2.2), the LU

TABLE 2.1
Computational costs due to m iterations.

Method	on one processor	on s processors
(2.3) with $L = 0$	$msd(C_f + 2d + 2s)$	$md(C_f + 2d + 2s)$
(2.3) with $L \neq 0$: LJ version	$msd(C_f + 4d + 3s)$	$md(C_f + 4ds + s^2)$
(2.4) with $L \neq 0$: LF version	$msd(C_f + 2d + 3s)$	$md(sC_f + 2ds + s^2)$
(2.5) transformed LJ version	$msd(C_f + 2d + 4s)$	$md(C_f + 2d + 4s)$

decompositions and the components of the residue $R(Y^{(j)})$ occurring in (2.3) and (2.4) can be evaluated in parallel. The schemes (2.3) and (2.4) will be called *parallel, triangularly implicit, iterated* methods.

In the case where (1.1) is an RK method, we shall refer to such methods as a PTIRK method and to distinguish them, we shall speak of the *LJ* and *LF versions*. In the case of (2.3), a further degree of parallelism is obtained by using the Butcher similarity transformation. This enables us to eliminate the triangularly implicit term $(L \otimes hJ)\Delta Y^{(j+1)}$ and leads to

$$(2.5) \qquad (I - D \otimes hJ)\Delta X^{(j+1)} = -(Q^{-1} \otimes I)R(Y^{(j)}),$$
$$Y^{(j+1)} = Y^{(j)} + (Q \otimes I)\Delta X^{(j+1)}, \quad BQ = QD.$$

In addition to the parallelism already present in (2.3) and (2.4), the scheme (2.5) also allows that in each iteration the s forward–backward substitutions can be done in parallel. Since the schemes (2.3) and (2.5) are algebraically identical, we shall call (2.5) the *transformed LJ version*.

Finally, we compare the computational costs of the various iteration schemes. These costs consists of two contributions, respectively, due to Jacobian updates and due to the successive iterations. In all schemes, the number of flops per step originating from the Jacobian updates is given by

$$C_1 = \frac{sd^2}{\nu}\left(\frac{1}{s}C_J + \frac{2}{3}d + 1\right),$$

where ν denotes the averaged number of steps during which the Jacobian and the LU decomposition are kept constant, and C_J denotes the average numbers of flops for computing one entry of J. The contribution C_1 is perfectly parallelizable and can be reduced effectively by a factor s on s processors. The contribution due to m (say) iterations are summarized in Table 2.1. In this table, C_f denotes the average numbers of flops for computing one component of f. Evidently, on a parallel computer, the methods (2.3) with $L = 0$ and (2.5) are the less expensive ones.

3. Convergence of the iteration process. In order to analyze convergence, we define the iteration error $\epsilon^{(j)} = Y^{(j)} - Y$, and we write the LJ and LF versions (2.3) and (2.4) in the respective forms

$$(3.1) \qquad (I - B \otimes hJ)(\epsilon^{(j+1)} - \epsilon^{(j)}) = -\epsilon^{(j)} + h(A \otimes I)(F(Y + \epsilon^{(j)}) - F(Y)),$$
$$(I - D \otimes hJ)(\epsilon^{(j+1)} - \epsilon^{(j)}) = -\epsilon^{(j)} + h((A - L) \otimes I)(F(Y + \epsilon^{(j)}) - F(Y))$$
$$+h(L \otimes I)(F(Y + \epsilon^{(j+1)}) - F(Y)).$$

The components of $F(Y + \epsilon) - F(Y)$ can be expanded according to $J_i\epsilon_i + \mathcal{O}(\epsilon_i^2)$, where J_i is the Jacobian matrix of the right-hand side function at Y_i. Assuming that

J is nonsingular, we may define the block-diagonal matrix ΔJ of which the diagonal blocks are given by $J^{-1}\Delta J_i = J^{-1}(J_i - J)$ to obtain

$$F(Y + \epsilon^{(j)}) - F(Y) = (I \otimes J)\epsilon^{(j)} + (I \otimes J)\Delta J\epsilon^{(j)} + \mathcal{O}((\epsilon^{(j)})^2).$$

Ignoring the second-order terms (first-order convergence analysis), the error recursions for the LJ and LF versions can be represented in the forms

$$\epsilon^{(j+1)} = M(I + P\Delta J)\epsilon^{(j)}, \tag{3.2}$$

$$\epsilon^{(j+1)} = (I - N\Delta J)^{-1}M(I + Q\Delta J)\epsilon^{(j)}, \tag{3.3}$$

where

$$M := (I - B \otimes hJ)^{-1}((A - B) \otimes hJ),$$
$$N := (I - B \otimes hJ)^{-1}(L \otimes hJ),$$
$$P := (A - B)^{-1}A \otimes I, \quad Q := (A - B)^{-1}(A - L) \otimes I.$$

If we ignore ΔJ (linear convergence analysis), then the error recursions of both versions are characterized by the matrix M. However, if ΔJ cannot be neglected, then the error recursions may behave quite differently. For example, as $h \to 0$, then we have

$$\epsilon^{(j+1)} \approx h((A - B) \otimes J + (A \otimes J)\Delta J)\epsilon^{(j)}, \tag{3.4}$$

$$\epsilon^{(j+1)} \approx h((A - B) \otimes J + ((A - L) \otimes J)\Delta J)\epsilon^{(j)}. \tag{3.5}$$

Since the strictly lower triangular blocks of the amplification matrices in (3.4) and (3.5) differ by the matrices $hL_{ij}\Delta J_j$, the convergence behavior may differ considerably and is highly problem dependent. In the remainder of this paper, we shall focus on the matrix M.

3.1. Rate of convergence. In order to select a suitable matrix B, we consider the convergence of the individual error components corresponding to the eigenvalues λ of J. From (3.1) it follows that these error components are amplified by the matrix Z defined by

$$Z = Z(z) = z(I - zB)^{-1}(A - B), \quad z := h\lambda.$$

Z will be called the *amplification matrix associated with* M. A measure for the rate of convergence of the individual error components is defined by the (averaged) amplification factors

$$\rho_j(z) := \sqrt[j]{\|Z(z)^j\|_\infty}, \tag{3.6}$$

where $\|\cdot\|_\infty$ denotes the maximum norm. Note that $\rho_\infty(z) = \rho(Z(z))$, $\rho(\cdot)$ being the spectral radius function. For the test equation $y' = \lambda y$, the value of $\rho_j(z)$ may be interpreted as the averaged factor by which the iteration error corresponding to $z = h\lambda$ is reduced in each iteration, until the corrector solution is reached. For more general problems, we have to deal with $\rho_j(z)$ where z runs through the spectrum of hJ.

The amplification factor at $z = \infty$ will be called the *stiff* amplification factor. In the neighborhood of the origin we may write

$$\rho_j(z) = |z|\sqrt[j]{\|(A - B)^j\|_\infty} + \mathcal{O}(z^2) =: \tilde{\rho}_j(z) + \mathcal{O}(z^2) \text{ as } z \to 0. \tag{3.7}$$

The quantity $\widetilde{\rho}_j(z)$ will be called the *nonstiff* amplification factor. Furthermore, we denote the maximal amplification factor in the left-hand plane by ρ_j^*. Of course, ρ_j^* refers to the worst case situation, but it serves as an indicator for the robustness of the method.

3.2. Iteration strategies. In this section, we discuss the choice of the free matrix $B = L + D$ in the iteration schemes (2.3) and (2.4). We first briefly review the *diagonal* iteration strategy of [15], i.e., $L = 0$, and then we focus on the *triangular* iteration strategy, where L is allowed to be an arbitrary strictly (lower) *triangular* matrix. A nonvanishing matrix L enables us to reduce the norm of Z^j considerably.

The reason for restricting B to the class of triangular matrices is that we have direct control on the eigenvalues of B. As a consequence, suitable matrices B can be constructed without performing a many-parameter search as was carried out in [15]. Since our main source of correctors is the class of RK methods which usually possess a dominant *lower* triangular part, B is also assumed to be *lower* triangular. (Recall that ideally B should equal A.) For both the diagonal iteration and the triangular iteration approach, the matrices B, $Z_0 := A - B$, and $Z_\infty := I - B^{-1}A$ associated with a number of classical RK methods are specified in the appendix to this paper.

3.2.1. Diagonal iteration. The diagonal iteration strategy is characterized by a diagonal matrix B with positive diagonal entries. In this strategy, it was found for a large number of classical RK correctors that small iteration error amplification factors for the stiff error components are crucial for a satisfactory overall convergence [15]. This is due to an order reduction effect common in stiff situations and can be explained by considering the error recursions (3.2) and (3.3). Due to the "Jacobian-defect" matrix ΔJ, the stiff error components are not damped as strongly as the nonstiff error components. Therefore, we determined in [15] the diagonal matrix $B = D$ such that Z_∞ has a minimal spectral radius; that is, the *asymptotic* value $\rho_\infty(\infty) = \rho(Z_\infty)$ of the stiff amplification factor is minimized. In [15] this was achieved by a multiparameter search over the diagonal entries of D. For a large number of collocation-based RK correctors, it turned out that the spectral radius $\rho(Z_\infty)$ of $Z_\infty = I - B^{-1}A$ is extremely small. In fact, we conjecture that for collocation-based RK correctors, there exist matrices D with positive diagonal entries for which $\rho(Z_\infty)$ actually vanishes. This suggests an alternative construction of the matrix $B = D$. Writing down the characteristic equation for Z_∞ and imposing the condition that this equation has only zero roots, we arrive at a (nonlinear) system for the entries d_{ii} of D. If this system can be solved for positive d_{ii}, $i = 1, \ldots, s$, then we have found an optimal matrix D. It has been verified for the Radau IIA correctors with $s \leq 8$ that such optimal matrices D do exist (see [18]). Notice that a zero spectral radius $\rho(Z_\infty)$ implies that Z_∞^j vanishes for $j \geq s$. (This can be seen by considering the Schur decomposition $Z_\infty = QTQ^{-1}$ with Q orthogonal and T strictly lower triangular.)

If the matrices D are obtained by a numerical search as in [15], then they will always give rise to a small but yet nonzero $\rho(Z_\infty)$. Nevertheless, for both the *nonstiff* and the *highly stiff* error components, the generated PDIRK methods show a satisfactory convergence rate for larger values of j. On the other hand, it also turns out that for the higher-order methods, there may be regions in the z-plane where $\rho_j(z)$ exceeds one for small j, so that initially error components corresponding to points lying in such regions will diverge [16, Table 3.2b]. The reason for this behavior is the "abnormality" of the matrix Z. In particular, for larger values of $|z|$, i.e., for the stiff error components, the matrix $Z(z)$ may differ considerably from a normal matrix. To be more precise, let the departure from normality of the matrix Z be defined by

$\Delta^2(Z) := \|Z\|_F^2 - \|\zeta(Z)\|_2^2$, where $\zeta(Z)$ denotes the vector of eigenvalues of Z and $\|\cdot\|_F$ and $\|\cdot\|_2$, respectively, denote the Frobenius matrix norm and the Euclidean vector norm (see, e.g., [12, p. 336]). By considering plots of $\Delta^2(Z)$ as a function of $|z|$ with $\arg(z)$ constant, we found that in the left-hand half-plane $\Delta^2(Z)$ monotonically increases from 0 to values greater than 20. This situation is particularly unfortunate if we want to apply the step-parallel iteration approach mentioned in section 1. In such an approach, it is crucial that in the whole left-hand half-plane the amplification factor is less than one right from the beginning.

3.2.2. Triangular iteration. In the triangular iteration strategy we choose B lower triangular with positive diagonal entries such that Z_∞ is *strictly upper triangular*. As a consequence, we have a zero stiff amplification factor for $j \geq s$. For the construction of such a matrix B we use the LU decomposition of A. Let $A = T_L T_U$ with T_L lower triangular and T_U unit upper triangular (Crout decomposition), and define $B = T_L$. Since $Z_\infty = I - B^{-1}A$, we immediately obtain the strictly upper triangular matrix $Z_\infty = I - T_U$. The following lemma provides an explicit criterion for the positiveness of the diagonal entries of $B = T_L$.

LEMMA 3.1. *Let A, L, D, and U be $s \times s$ matrices such that $A = LDU$ with L unit lower triangular, D diagonal, and U unit upper triangular, and let A_k denote the $k \times k$ principal submatrix of A. Then D has diagonal entries given by*

$$d_k = \frac{\det(A_k)}{\det(A_{k-1})}, \quad k = 1, \ldots, s, \tag{3.8}$$

where $\det(A_0) := 1$ *and* $\det(A_1) := a_{11}$.

Proof. Let A_i be decomposed according to $A_k = L_k D_k U_k$ with L_k unit lower triangular, D_k diagonal, and U_k unit upper triangular. Then

$$\det(A_k) = \det(L_k)\det(D_k)\det(U_k) = \det(D_k).$$

Since the first $k-1$ pivots in the Gaussian elimination process do not depend on the entries a_{ij} with $i \geq k$ and $j \geq k$, it follows therefore that the diagonal entries d_{ik} of D_k are defined by $d_{ik} = d_i$. Hence, $\det(D_k) = \det(D_{k-1})d_k$, which is equivalent with (3.8). □

From this lemma it follows that the diagonal entries of the matrix B defined above are given by (3.8), so that they are all positive, if all values $\det(A_k)$, $k = 1, \ldots, s$, are positive. In the following, we restrict our considerations to collocation methods with distinct abscissas c_i. Such methods are generated by matrices A of the form (see, e.g., [14, p. 82])

$$\begin{aligned} &A = CVRV^{-1}, \quad C = \operatorname{diag}(c), \quad R = \operatorname{diag}(r), \\ &c = (c_i), \qquad\quad r = (i^{-1}), \qquad V = (e\ c\ c^2\ \ldots\ c^{s-1}), \end{aligned} \tag{3.9}$$

where $i = 1, \ldots, s$ and e is the vector with unit entries.

THEOREM 3.1. *If A results from a collocation method with positive, distinct abscissas ordered according to $0 < c_1 < c_2 < \cdots < c_s$, then the following results hold:*

1. *The values of* $\det(A_1)$ *and* $\det(A_s)$ *are positive for all s.*
2. *Let V and R be partitioned according to* (3.10), *where V_k and R_k denote the $k \times k$ principal submatrices of the matrices V and R defined in* (3.9). *Then, for $1 < k < s$,* $\det(A_k)$ *is positive if*

$$q_k := \det(V_k R_k - PSW^{-1}Q) > 0, \quad V = \begin{bmatrix} V_k & P \\ Q & W \end{bmatrix}, \quad R = \begin{bmatrix} R_k & 0 \\ 0 & S \end{bmatrix}. \tag{3.10}$$

Proof.

1. For collocation methods we have that

$$a_{11} = \int_0^{c_1} \frac{c_2 - t}{c_2 - c_1} \frac{c_3 - t}{c_3 - c_1} \cdots \frac{c_s - t}{c_s - c_1} dt.$$

From the condition on the collocation points it is immediate that $\det(A_1) = a_{11} >$ and from (3.9) it follows that $\det(A) = \det(C)\det(VRV^{-1}) = \det(C)\det(R) > 0$.

2. By means of (3.9) it is easily verified that A_k can be presented in the for

$$A_k = C_k(V_k R_k - PSW^{-1}Q)(V_k - PW^{-1}Q)^{-1}, \tag{3.11}$$

where C_k is the $k \times k$ principal submatrix of C and V_k, R_k, P, S, W, and Q specified in (3.10). We now prove that $\det(V_k - PW^{-1}Q)$ is positive by consider the factorizations

$$\begin{bmatrix} V_k - PW^{-1}Q & 0 \\ Q & W \end{bmatrix} = \begin{bmatrix} I & -PW^{-1} \\ 0 & I \end{bmatrix} V,$$

$$\begin{bmatrix} V_k - PW^{-1}Q & 0 \\ Q & W \end{bmatrix} = \begin{bmatrix} V_k - PW^{-1}Q & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ Q & W \end{bmatrix}.$$

From these two relations it follows that $\det(V) = \det(V_k - PW^{-1}Q)\det(W)$. Since is a Vandermonde matrix and W is a row-scaled Vandermonde matrix, we conclu that both V and W have a positive determinant. Thus, $\det(V_k - PW^{-1}Q) > 0$, a by virtue of (3.11), it follows that $\det(A_k)$ is positive whenever the quantity q_k defi in (3.10) is positive. □

Theorem 3.1 directly implies the positiveness of the diagonal entries of $B = T_L$ all two-stage collocation methods. For higher-stage methods, it provides the relativ simple criterion $q_k > 0$, $1 < k < s$, for verifying the condition $\det(A_k) > 0$. conjecture that the condition $\det(A_k) > 0$, $1 \le k \le s$, is true for all s, but so we are not able to prove it. Instead, we verified the correctness of this conjecture $s \le 6$. An easy way of verifying the conditions $q_k > 0$ replaces the abscissas c_i in by $c_i = p_i + p_{i-1} + \cdots + p_1$ and expresses q_k as a rational function of the s paramet p_i. For $s \le 6$, it turns out that all coefficients in this rational expression are positi Since the parameters p_i are all positive (because $p_i := c_i - c_{i-1}$ with $c_0 := 0$), t implies that q_k is positive.

Example 3.1. For $s = 3$, we have to prove that $q_2 > 0$. A straightforw calculation yields

$$q_2 = \frac{3p_2p_3^2 + 4p_2^2p_3 + 2p_1p_2p_3 + p_1p_2^2 + p_2^3}{6(p_1 + p_2 + p_3)^2},$$

which is obviously positive. □

Summarizing we conclude that, unlike the diagonal approach, the triangular proach provides an extremely simple construction of the matrix B and an expli criterion for checking the positiveness of its diagonal entries. Moreover, it turns that for larger values of $|z|$ the departure from normality $\Delta^2(Z) := \|Z\|_F^2 - \|\zeta(Z$ is considerably reduced. This can be explained by the fact that the magnitude of entries of the matrix Z (and hence $\|Z\|_F^2$) can be made much smaller by a triangu matrix B than by a diagonal matrix B. (Recall that Z contains the factor $A -$ Plots show that $\Delta^2(Z)$ monotonically increases from zero at the origin to values l than 0.4 at infinity, resulting in amplification factors that are less than one in whole left half-plane for all j. This will be quantified in the following subsection.

TABLE 3.1
Amplification factors $\widetilde{\rho}_j$, $\rho_j(\infty)$, and ρ_j^*.

	Method	Gauss	Lobatto IIIA		Radau IIA		
		$s=2$	$s=2$	$s=3$	$s=2$	$s=3$	$s=4$
$\widetilde{\rho}_1(z)$	PDIRK	$0.79\|z\|$	$0.89\|z\|$	$0.91\|z\|$	$1.15\|z\|$	$1.15\|z\|$	$1.10\|z\|$
	PTIRK	$0.08\|z\|$	$0.08\|z\|$	$0.13\|z\|$	$0.15\|z\|$	$0.21\|z\|$	$0.25\|z\|$
$\widetilde{\rho}_2(z)$	PDIRK	$0.36\|z\|$	$0.31\|z\|$	$0.32\|z\|$	$0.52\|z\|$	$0.44\|z\|$	$0.52\|z\|$
	PTIRK	$0.08\|z\|$	$0.08\|z\|$	$0.13\|z\|$	$0.15\|z\|$	$0.20\|z\|$	$0.22\|z\|$
$\widetilde{\rho}_3(z)$	PDIRK	$0.45\|z\|$	$0.21\|z\|$	$0.30\|z\|$	$0.40\|z\|$	$0.34\|z\|$	$0.25\|z\|$
	PTIRK	$0.08\|z\|$	$0.08\|z\|$	$0.12\|z\|$	$0.15\|z\|$	$0.20\|z\|$	$0.20\|z\|$
$\rho_1(\infty)$	PDIRK	1.58	2.29	5.62	1.78	4.17	4.68
	PTIRK	0.15	0.13	0.23	0.20	0.46	0.68
$\rho_2(\infty)$	PDIRK	0	0	3.09	0	1.82	3.31
	PTIRK	0	0	0.17	0	0.26	0.47
$\rho_3(\infty)$	PDIRK	0	0	0	0	0	2.09
	PTIRK	0	0	0	0	0	0.30
ρ_1^*	PDIRK	1.58	2.29	5.62	1.78	4.17	4.68
	PTIRK	0.15	0.13	0.23	0.20	0.46	0.68
ρ_2^*	PDIRK	0.59	0.58	3.09	0.63	1.82	3.31
	PTIRK	0.14	0.14	0.33	0.18	0.40	0.58
ρ_3^*	PDIRK	0.45	0.39	1.35	0.47	1.02	2.09
	PTIRK	0.14	0.14	0.32	0.18	0.39	0.55
ρ_∞^*	PDIRK	0.25	0.17	0.45	0.26	0.40	0.52
	PTIRK	0.14	0.14	0.30	0.18	0.37	0.50

3.3. Comparison of amplification factors. For a number of well-known RK correctors, we compare the amplification factors defined by (3.6) associated with the diagonal approach (PDIRK method) and the triangular approach (PTIRK method). For $j = 1, 2, 3$, Table 3.1 presents the *nonstiff* amplification factor $\widetilde{\rho}_j(z)$ as defined in (3.7), the *stiff* amplification factor $\rho_j(\infty)$, and the maximal amplification factor ρ_j^*. These figures indicate that, initially, the PTIRK strategy converges considerably faster than the PDIRK strategy. Hence, it should be a sound starting point for step-parallel applications. This will be subject of future research.

4. Numerical illustration. In this section, we compare the diagonally implicit iteration (PDIRK) strategy with the triangularly implicit iteration (PTIRK) strategy. In all experiments, we used the four-stage Radau IIA corrector with constant stepsizes and a Jacobian update in each step. If necessary, the initial condition in the problems below is adapted such that the integration starts outside the transient phase, enabling us to use constant steps. Two predictors were tested, the simple last step value (LSV) predictor $Y^{(0)} = (e_s^{\mathrm{T}} \otimes I)Y$ and the extrapolation (EPL) predictor $Y^{(0)} = (E \otimes I)Y$. Here, Y denotes the stage vector from the preceding step, e_s is the sth unit vector, and E is the extrapolation matrix. In the following two subsections we compare the accuracy and the CPU time on a four-processor Cray C98/4256 of the PDIRK and PTIRK methods.

4.1. Accuracy tests. In the tables of results, the LF and LJ versions (2.3) and (2.4) of the PTIRK method are indicated by PTIRK(LJ) and PTIRK(LF). For a given number of iterations m, the tables of results below present the minimal number of correct digits cd of the components of the numerical solution at the end point $t = T$ of the integration interval; that is, at the end point the absolute errors are written as 10^{-cd}. Our first example (Tables 4.1a and 4.1b) is provided by a problem of Schäfer, called the HIRES problem in [14, p. 157]. It was proposed in Gottwald [13] as a test

TABLE 4.1a
LSV predictor: HIRES problem of Schäfer.

Method	h	$m=1$	$m=2$	$m=3$	$m=4$	...	$m=10$
PDIRK	15	*	*	*	4.3	...	6.5
PTIRK(LJ)	15	3.4	3.5	3.8	4.2	...	6.3
PTIRK(LF)	15	3.1	4.0	3.9	4.1	...	5.6
PDIRK	7.5	*	*	*	5.4	...	7.7
PTIRK(LJ)	7.5	4.0	4.2	4.7	5.1	...	8.3
PTIRK(LF)	7.5	3.3	4.4	4.7	5.3	...	7.0

TABLE 4.1b
EPL predictor: HIRES problem of Schäfer.

Method	h	$m=1$	$m=2$	$m=3$	$m=4$	...	$m=10$
PDIRK	15	*	*	*	*	...	6.4
PTIRK(LJ)	15	*	3.0	4.8	5.1	...	7.3
PDIRK	7.5	*	*	*	4.1	...	8.8
PTIRK(LJ)	7.5	*	2.5	6.1	6.6	...	9.0

TABLE 4.2a
LSV predictor: Chemical reaction problem of Gear.

Method	h	$m=1$	$m=2$	$m=3$	$m=4$	...	$m=10$
PDIRK	50	1.4	2.2	2.6	2.9	...	5.2
PTIRK(LJ)	50	2.3	2.7	3.5	4.3	...	7.7
PTIRK(LF)	50	1.8	2.9	3.9	3.0	...	3.3
PDIRK	25	1.8	2.9	3.4	3.6	...	7.3
PTIRK(LJ)	25	2.3	3.6	4.2	5.3	...	9.8
PTIRK(LF)	25	2.1	4.3	4.4	4.6	...	6.4

TABLE 4.2b
EPL predictor: Chemical reaction problem of Gear.

Method	h	$m=1$	$m=2$	$m=3$	$m=4$	...	$m=10$
PDIRK	25	2.4	2.8	3.2	3.6	...	7.4
PTIRK(LJ)	25	2.9	3.7	4.3	5.6	...	9.8

TABLE 4.3a
LSV predictor: ATMOS20 problem of Verwer.

Method	h	$m=1$	$m=2$	$m=3$	$m=4$	...	$m=10$
PDIRK	11	2.7	2.1	2.8	5.4	...	9.8
PTIRK(LJ)	11	3.3	5.0	6.1	6.7	...	11.0
PTIRK(LF)	11	3.4	4.9	7.0	6.8	...	8.7
PDIRK	5.5	1.3	*	*	6.5	...	11.2
PTIRK(LJ)	5.5	3.7	5.6	7.0	7.7	...	12.2
PTIRK(LF)	5.5	3.7	5.5	7.6	8.3	...	11.5
PDIRK	2.25	*	*	*	7.5	...	12.2
PTIRK(LJ)	2.25	4.0	6.2	7.8	8.6	...	12.6
PTIRK(LF)	2.25	4.0	6.2	8.2	10.0	...	12.1

problem and consists of eight mildly stiff equations on the interval $5 \leq t \leq 305$. (It included in the CWI test set [22].) The second test problem (Tables 4.2a and 4.2b) a set of three chemical reaction equations originating from Gear [10] on the interv [1,51] and is included in the test set of Enright, Hull, and Lindberg [9]. The ATMOS

TABLE 4.3b
EPL predictor: ATMOS20 problem of Verwer.

Method	h	$m=1$	$m=2$	$m=3$	$m=4$	...	$m=10$
PDIRK	11	1.8	2.6	2.1	5.4	...	10.0
PTIRK(LJ)	11	2.0	3.7	6.3	7.0	...	10.9
PDIRK	5.5	*	*	*	6.4	...	11.8
PTIRK(LJ)	5.5	*	4.2	7.4	8.2	...	12.2
PDIRK	2.25	*	*	*	8.2	...	12.7
PTIRK(LJ)	2.25	*	*	8.6	9.4	...	12.7

TABLE 4.4
Ring modulator of Horneber.

Method		$m=2$	$m=3$	$m=4$	...	$m=10$
PDIRK	$cd=$	*	*	4.6	...	8.5
	CPU(1) =	*	*	26.2	...	51.9
	CPU(4) =	*	*	8.2	...	16.7
PTIRK(LJ)	$cd=$	5.7	7.7	8.3	...	8.5
	CPU(1) =	20.1	23.1	28.1	...	56.6
	CPU(4) =	6.1	7.0	8.8	...	18.3
RADAU5	$cd=$	5.8	6.5	7.2		
	CPU(1) =	14.0	17.5	21.9		
	$10^7 h =$	1.25	1.0	0.8		

problem is our third test problem (Tables 4.3a and 4.3b). It is a system of 20 stiff nonlinear ODEs originating from an air pollution model used by Verwer [23] and included in the CWI test set [22]. We solved this system in the integration interval [5, 60].

The tables of results clearly show for both predictors the superiority of the PTIRK strategy in the first few iterations. For large numbers of iterations, PDIRK and PTIRK(LJ) are better than PTIRK(LF).

4.2. Cray C98/4256 tests. We applied the PDIRK and PTIRK(LJ) methods on a four-processor Cray C98/4256 to the Ring modulator of Horneber. This problem consists of 15 highly stiff differential equations on the interval [0,0.001]. (For details, see the CWI test set [22].) PDIRK and PTIRK(LJ) were applied with the EPL predictor and stepsize $h = 1.25 \cdot 10^{-7}$. We compiled the codes with `cf77` using the flags `-dp`, `-Zp`, `-Wu-p`, and `-Wd-dj`. The environmental variable `NCPUS` had the value 4. For the meaning of these settings, we refer to Cray Research Inc., CF77 Commands and Directives, SR-3771, 6.0 edition, 1994. Table 4.4 lists the *cd*-values and the CPU(1) and CPU(4) timings (in sec.) required on one and four processors, respectively. For the CPU(1) timings we used the internal function `SECOND` and the CPU(4)-values were obtained by using the Cray tool `ATExpert`. These figures again show that PTIRK(LJ) is much more accurate than PDIRK, while it is hardly more expensive than PDIRK (less than 10%). Since earlier experiments on a four-processor ALLIANT FX/4 indicated that the PDIRK-based code PSODE is in general twice as efficient as LSODE and RADAU5 [21, p. 12], we expect that a PTIRK-based code should be at least twice as efficient as LSODE and RADAU5. The present version of the PTIRK code does not yet contain a sufficiently tested stepsize and iteration-stopping strategy. Therefore it is not yet possible to compare it with codes like LSODE and RADAU5. Nevertheless, in order to have some indication how PTIRK performs in a parallel environment, we applied RADAU5 with the same integration strategy as our present PTIRK code, that

is, with constant stepsizes (without step rejections) and with a Jacobian update in each step. In order to generate a range of cd-values, we ran RADAU5 with a few different stepsizes. The cd-values and CPU(1) timings obtained are also listed in Table 4.4.

We finish this paper with the following conclusions:

1. For a relatively difficult problem as provided by the Ring modulator, the PTIRK code on a Cray in four-processor mode shows a speed-up ranging from > 2.4 to > 3.1 with respect to RADAU5 on a Cray in one-processor mode.

2. It is not expected that a parallel implementation of RADAU5 is as efficient as PTIRK, because the intrinsic parallelism of PTIRK is much larger than that of RADAU5. For example, the effective LU costs and forward–backward substitutions are four times as expensive, due to the fact that the eigenvalues of the Radau IIA matrix are not all real, so that the Butcher transformation used in RADAU5 decouples the $3d$-dimensional system into one real and one *complex* system of dimension d. The LU decompositions and the forward–backward substitutions associated with these systems can be done concurrently. However, complex arithmetic is about four times as expensive as real arithmetic, which explains the factor 4 mentioned above.

A. Parameter matrices. For a number of RK methods, we have computed the matrices $B = L + D$ according to the procedure outlined in subsections 3.2.1 and 3.2.2, together with the amplification matrices Z_0 and Z_∞.

A.1. PDIRK strategy.

$$Z(z) = z(I - zD)^{-1}(A - D), \quad Z_0 = A - D, \quad Z_\infty = I - D^{-1}A.$$

A.1.1. Radau IIA.

$s = 2$:

$$D = \begin{bmatrix} 0.2584 & 0 \\ 0 & 0.6449 \end{bmatrix}, \qquad Z_0 = \begin{bmatrix} 0.1582 & -0.0833 \\ 0.7500 & -0.3949 \end{bmatrix},$$

$$Z_\infty = \begin{bmatrix} -0.6124 & 0.3225 \\ -1.1629 & 0.6124 \end{bmatrix}.$$

$s = 3$:

$$D = \begin{bmatrix} 0.3204 & 0 & 0 \\ 0 & 0.1400 & 0 \\ 0 & 0 & 0.3717 \end{bmatrix}, \quad Z_0 = \begin{bmatrix} -0.1236 & -0.0655 & 0.0238 \\ 0.3944 & 0.1521 & -0.0415 \\ 0.3764 & 0.5125 & -0.2606 \end{bmatrix},$$

$$Z_\infty = \begin{bmatrix} 0.3857 & 0.2045 & -0.0742 \\ -2.8179 & -1.0867 & 0.2968 \\ -1.0127 & -1.3789 & 0.7011 \end{bmatrix}.$$

$s = 4$:

$$D = \begin{bmatrix} 0.3205 & 0 & 0 & 0 \\ 0 & 0.0892 & 0 & 0 \\ 0 & 0 & 0.1817 & 0 \\ 0 & 0 & 0 & 0.2334 \end{bmatrix}, \quad Z_0 = \begin{bmatrix} -0.2075 & -0.0403 & 0.0258 & -0.0099 \\ 0.2344 & 0.1177 & -0.0479 & 0.0160 \\ 0.2167 & 0.4061 & 0.0073 & -0.0242 \\ 0.2205 & 0.3882 & 0.3288 & -0.1709 \end{bmatrix},$$

$$Z_\infty = \begin{bmatrix} 0.6474 & 0.1258 & -0.0805 & 0.0309 \\ -2.6290 & -1.3206 & 0.5368 & -0.1800 \\ -1.1923 & -2.2346 & -0.0402 & 0.1331 \\ -0.9447 & -1.6635 & -1.4092 & 0.7322 \end{bmatrix}.$$

A.1.2. Lobatto IIIA.

$s = 2$:

$$D = \begin{bmatrix} 0.2113 & 0 \\ 0 & 0.3943 \end{bmatrix}, \qquad Z_0 = \begin{bmatrix} 0.1220 & -0.0417 \\ 0.6667 & -0.2277 \end{bmatrix},$$

$$Z_\infty = \begin{bmatrix} -0.5774 & 0.1972 \\ -1.6906 & 0.5774 \end{bmatrix}.$$

$s = 3$:

$$D = \begin{bmatrix} 0.4802 & 0 & 0 \\ 0 & 0.1094 & 0 \\ 0 & 0 & 0.1604 \end{bmatrix}, \quad Z_0 = \begin{bmatrix} -0.2905 & -0.0339 & 0.0103 \\ 0.4506 & 0.1175 & -0.0270 \\ 0.4167 & 0.4167 & -0.0770 \end{bmatrix},$$

$$Z_\infty = \begin{bmatrix} 0.6049 & 0.0706 & -0.0215 \\ -4.1179 & -1.0743 & 0.2465 \\ -2.5981 & -2.5981 & 0.4804 \end{bmatrix}.$$

A.1.3. Gauss.

$s = 2$:

$$D = \begin{bmatrix} 0.1667 & 0 \\ 0 & 0.5000 \end{bmatrix}, \quad Z_0 = \begin{bmatrix} 0.0833 & -0.0387 \\ 0.5387 & -0.2500 \end{bmatrix},$$

$$Z_\infty = \begin{bmatrix} -0.5000 & 0.2321 \\ -1.0774 & 0.5000 \end{bmatrix}.$$

A.2. PTIRK strategy.

$$Z(z) = z(I - zB)^{-1}(A - B), \quad Z_0 = A - B, \quad Z_\infty = I - B^{-1}A.$$

A.2.1. Radau IIA.

$s = 2$:

$$B = \begin{bmatrix} 0.4167 & 0 \\ 0.7500 & 0.4000 \end{bmatrix}, \qquad Z_0 = \begin{bmatrix} 0 & -0.0833 \\ 0 & -0.1500 \end{bmatrix},$$

$$Z_\infty = \begin{bmatrix} 0 & 0.2000 \\ 0 & 0 \end{bmatrix}.$$

$s = 3$:

$$B = \begin{bmatrix} 0.1968 & 0 & 0 \\ 0.3944 & 0.4234 & 0 \\ 0.3764 & 0.6378 & 0.2000 \end{bmatrix}, \qquad Z_0 = \begin{bmatrix} 0 & -0.0655 & 0.0238 \\ 0 & -0.1313 & -0.0415 \\ 0 & -0.1253 & -0.0889 \end{bmatrix},$$

$$Z_\infty = \begin{bmatrix} 0 & 0.3330 & -0.1208 \\ 0 & 0 & 0.2106 \\ 0 & 0 & 0 \end{bmatrix}.$$

$s = 4$:

$$B = \begin{bmatrix} 0.1130 & 0 & 0 & 0 \\ 0.2344 & 0.2905 & 0 & 0 \\ 0.2167 & 0.4834 & 0.3083 & 0 \\ 0.2205 & 0.4668 & 0.4414 & 0.1176 \end{bmatrix}, \quad Z_0 = \begin{bmatrix} 0 & -0.0403 & 0.0258 & -0.0099 \\ 0 & -0.0836 & -0.0479 & 0.0160 \\ 0 & -0.0773 & -0.1192 & -0.0242 \\ 0 & -0.0786 & -0.1126 & -0.0551 \end{bmatrix},$$

$$Z_\infty = \begin{bmatrix} 0 & 0.3567 & -0.2283 & 0.0877 \\ 0 & 0 & 0.3490 & -0.1260 \\ 0 & 0 & 0 & 0.2144 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

A.2.2. Lobatto IIIA.

$$s = 2: \quad B = \begin{bmatrix} 0.3333 & 0 \\ 0.6667 & 0.2500 \end{bmatrix}, \quad Z_0 = \begin{bmatrix} 0 & -0.0417 \\ 0 & -0.0833 \end{bmatrix}, \quad Z_\infty = \begin{bmatrix} 0 & 0.1250 \\ 0 & 0 \end{bmatrix}.$$

$$s = 3: \quad B = \begin{bmatrix} 0.1897 & 0 & 0 \\ 0.4506 & 0.3075 & 0 \\ 0.4167 & 0.4911 & 0.1429 \end{bmatrix}, \quad Z_0 = \begin{bmatrix} 0 & -0.0339 & 0.0103 \\ 0 & -0.0805 & -0.0270 \\ 0 & -0.0745 & -0.0595 \end{bmatrix}, \quad Z_\infty = \begin{bmatrix} 0 & 0.1787 & -0.0543 \\ 0 & 0 & 0.1673 \\ 0 & 0 & 0 \end{bmatrix}.$$

A.2.3. Gauss.

$$s = 2: \quad B = \begin{bmatrix} 0.2500 & 0 \\ 0.5387 & 0.3333 \end{bmatrix}, \quad Z_0 = \begin{bmatrix} 0 & 0.0387 \\ 0 & 0.0833 \end{bmatrix}, \quad Z_\infty = \begin{bmatrix} 0 & 0.1547 \\ 0 & 0 \end{bmatrix}.$$

Acknowledgments. The authors are grateful to Dr. B. P. Sommeijer and to Dr. W. Hoffmann for the many valuable discussions on this paper.

REFERENCES

[1] A. BELLEN, *Parallelism across the steps for difference and differential equations*, Lecture Notes in Mathematics 1386, Springer-Verlag, Berlin, 1987, pp. 22–35.

[2] A. BELLEN, Z. JACKIEWICZ, AND M. ZENNARO, *Time-point relaxation Runge–Kutta methods for ordinary differential equations*, J. Comput. Appl. Math., 45 (1990), pp. 121–138.

[3] K. BURRAGE, *A special family of Runge–Kutta methods for solving stiff differential equations*, BIT, 18 (1978), pp. 373–383.

[4] K. BURRAGE, *Parallel methods for initial value problems*, Appl. Numer. Math., 11 (1993), pp. 5–26.

[5] K. BURRAGE, *The search for the Holy Grail, or: Predictor-corrector methods for solving ODEIVPs*, Appl. Numer. Math., 11 (1993), pp. 125–141.

[6] J. C. BUTCHER, *On the implementation of implicit Runge–Kutta methods*, BIT, 16 (1976), pp. 237–240.

[7] J. C. BUTCHER, *The Numerical Analysis of Ordinary Differential Equations, Runge–Kutta and General Linear Methods*, Wiley, New York, 1987.

[8] P. CHARTIER, *Parallelism in the Numerical Solutions of Initial Value Problems for ODEs and DAEs*, Ph.D. thesis, Université de Rennes I, France, 1993.

[9] W. H. ENRIGHT, T. E. HULL, AND B. LINDBERG, *Comparing numerical methods for stiff systems of ODEs*, BIT, 15 (1975), pp. 10–48.

[10] C. W. GEAR, *The automatic integration of stiff ordinary differential equations*, in Proc. IFIP Congress 1968, North–Holland, Amsterdam, 1969, pp. 187–193.

[11] C. W. GEAR AND X. XUHAI, *Parallelism across time in ODEs*, Appl. Numer. Math., 11 (1993), pp. 45–68.

[12] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., John Hopkins University Press, Baltimore, MD, London, 1989.

[13] B. A. GOTTWALD, *MISS - ein einfaches simulations-system für biologische und chemische prozesse*, EDV in Medizin und Biologie, 3 (1977), pp. 85–90.

[14] E. HAIRER AND G. WANNER, *Solving Ordinary Differential Equations* II: *Stiff and Differential-algebraic Problems*, Springer-Verlag, Berlin, 1991.

[15] P. J. VAN DER HOUWEN and B. P. SOMMEIJER, *Iterated Runge–Kutta methods on parallel computers*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 1000–1028.

[16] P. J. VAN DER HOUWEN, B. P. SOMMEIJER, AND W. A. VAN DER VEEN, *Parallelism across the steps in iterated Runge–Kutta methods for stiff initial value problems*, Numer. Algorithms, 8 (1994), pp. 293–312.

[17] P. J. VAN DER HOUWEN, B. P. SOMMEIJER, AND W. A. VAN DER VEEN, *Parallel iteration across the steps of high order Runge–Kutta methods for nonstiff initial value problems*, J. Comput. Appl. Math., 60 (1995), pp. 309–329.

[18] W. M. LIOEN, *On the Diagonal Approximation of Full Matrices*, Tech. Report NM-R9518, CWI, Amsterdam, the Netherlands, 1995, J. Comput. Appl. Math., to appear.

[19] W. L. MIRANKER AND W. LINIGER, *Parallel methods for the numerical integration of ordinary differential equations*, Math. Comp., 21 (1967), pp. 303–320.

[20] B. OREL, *Parallel Runge–Kutta methods with real eigenvalues*, Appl. Numer. Math., 11 (1993), pp. 241–250.

[21] B. P. SOMMEIJER, *Parallelism in the numerical integration of initial value problems*, CWI Tract 99, CWI, Amsterdam, the Netherlands, 1993.

[22] W. M. LIOEN, J. J. B. DE SWART, AND W. A. VAN DER VEEN, *Test set for IVP solvers*. Available via WWW at URL http://www.cwi.nl/cwi/projects/IVPtestset.shtml, 1996.

[23] J. G. VERWER, *Gauss-Seidel iteration for stiff ODEs from chemical kinetics*, SIAM J. Sci. Comput., 15 (1994), pp. 1243–1259.