SIAM J. SCI. STAT. COMPUT.
Vol. 12, No. 5, pp. 1000-1028, September 1991

© 1991 Society for Industrial and Applied Mathematics
002

ITERATED RUNGE-KUTTA METHODS ON PARALLEL COMPUTERS*

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

Abstract. This paper examines diagonally implicit iteration methods for solving implicit Runge-Kutta methods with high stage order on parallel computers. These iteration methods are such that after a finite number of m iterations, the iterated Runge-Kutta method belongs to the class of diagonally implicit Runge-Kutta methods (DIRK methods) using mk implicit stages where k is the number of stages of the generating implicit Runge-Kutta method (corrector method). However, a large number of the stages of this DIRK method can be computed in parallel, so that the number of stages that have to be computed sequentially is only m. The iteration parameters of the method are tuned in such a way that fast convergence to the stability characteristics of the corrector method is achieved. By means of numerical experiments it is also shown that the solution produced by the resulting iteration method converges rapidly to the corrector solution so that both stability and accuracy characteristics are comparable with those of the corrector. This implies that the reduced accuracy often shown when integrating stiff problems by means of DIRK methods already available in the literature (which is caused by a low stage order) is not shown by the DIRK methods developed in this paper, provided that the corrector method has a sufficiently high stage order.

Key words. diagonally implicit Runge-Kutta methods, parallelism, stability

AMS(MOS) subject classifications. 65M10, 65M20

C.R. classification. 5.17

1. Introduction.

1.1. Runge-Kutta methods. Suppose that we want to solve stiff initial-value problems for systems of first-order, ordinary differential equations (ODEs), i.e.,

$$\frac{d\mathbf{y}(t)}{dt}=\mathbf{f}(t,\mathbf{y}(t)),\qquad \mathbf{y}(t_0)=\mathbf{y}_0,\qquad \mathbf{y}:\mathbb{R}\to\mathbb{R}^d,\qquad \mathbf{f}:\mathbb{R}\times\mathbb{R}^d\to\mathbb{R}^d, \tag{1.1}$$

by means of a Runge-Kutta (RK) method. Then the stiffness of the problem requires that the RK method should be sufficiently stable, preferably A-stable, and therefore implicit. This leads us to fully implicit RK methods (IRK methods) in which the Butcher array,

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^T \end{array}, \tag{1.2}$$

has a full A matrix. Most widely used are the IRK methods based on Gaussian quadrature formulas (such as Gauss-Legendre, Lobatto, and Radau methods), which are known to be A-stable for any order of accuracy. However, the high degree of implicitness of these methods implies that solving the implicit relations is rather costly. In general, a k-stage IRK method (that is, $\mathbf{b}$ and $\mathbf{c}$ are k-dimensional vectors and A is a k-by-k matrix) requires in each step the solution of a system of dimension kd, so that the computational complexity is of order $(kd)^3$. This compares unfavourably with implicit linear multistep methods which require in each step the solution of a system of dimension d.

In order to reduce the computational labour involved when using implicit RK methods, various people have considered diagonally implicit RK methods (DIRK methods) possessing a lower triangular A matrix and therefore requiring (in general)

* Received by the editors January 16, 1990; accepted for publication (in revised form) September 12, 1990.
† Centre for Mathematics and Computer Science, Post Box 4079, 1009 AB Amsterdam, the Netherlands.

in each step the solution of k systems of dimension d. Hence, the computational complexity is now of order kd^3 instead of order $(kd)^3$. Unfortunately, the price we have to pay for the less expensive DIRK methods is a considerable drop in accuracy in many stiff problems. This is caused by the phenomenon of order reduction (cf., e.g., [21], [9], [11]) which reduces the observed order of RK methods to their stage order (or their stage order plus one). Most DIRK methods are particularly sensitive to order reduction because their stage order is only one or two, which is much smaller than for k-stage Gauss-Legendre, Lobatto IIIA and Radau IIA methods which have all stage order k.

An alternative for the DIRK methods are the singly implicit RK methods (SIRK methods) of Burrage [2] which possess a high stage order. By means of a transformation technique due to Butcher (see [5], [6]), these SIRK methods can be transformed into methods that are, like DIRK methods, only diagonally implicit. However, the additional transformations required in each step cause that the total costs per step are considerably higher than for DIRK methods.

Yet another possibility is the use of parallel processors. In this paper, we shall show that on parallel computers the fully implicit relations associated with IRK methods can be solved efficiently by using the highly parallelizable iteration methods of diagonally implicit type proposed in van der Houwen, Sommeijer, and Couzy [13]. This brings us back to using IRK methods as corrector method instead of using DIRK or SIRK methods. In particular, we shall concentrate on iterating IRK methods possessing high *stage* orders.

1.2. IRK methods with high stage orders. Most IRK methods are designed in such a way that they have a high order at the step points. However, as already remarked above, a high order at step points is often spoiled by order reduction, so that it seems more natural to look for IRK methods with as high a stage order as possible. In order to achieve this, we shall consider $(k+1)$-stage IRK methods of the type

$$\begin{array}{c|cc} 0 & 0 & \mathbf{0}^T \\ \mathbf{c} & \mathbf{a} & A \\ \hline & b_0 & \mathbf{b}^T \end{array}, \tag{1.3}$$

where b_0 is a scalar, $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ are k-dimensional vectors, and A is again a k-by-k matrix. IRK methods of this type have roughly the same computational complexity as the IRK methods of type (1.2), but they possess the additional parameter vector $\mathbf{a}$ which can be used for increasing the stage order. To see that (1.2) and (1.3) are (almost) equally expensive, let us assume (for simplicity of notation) that (1.1) is a scalar problem (i.e., $d=1$), and let us introduce the vectors

$$\mathbf{Y}_{n+1} := (y_{n,1}, \cdots, y_{n,k})^T, \qquad \mathbf{c} := (c_1, \cdots, c_k)^T,$$

where $y_{n,i}$ denotes a numerical approximation to the exact solution value $y(t_n + c_i h)$, h being the stepsize. Then we can write (1.3) in the form

$$\begin{aligned} &\mathbf{Y}_{n+1} - hAf(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_{n+1}) = \mathbf{e}y_n + h\mathbf{a}f(t_n, y_n), \\ &y_{n+1} = y_n + hb_0 f(t_n, y_n) + h\mathbf{b}^T f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_{n+1}). \end{aligned} \tag{1.3'}$$

Here, $\mathbf{e}$ is the vector with unit entries, and we used the convention that for any given vectors $\mathbf{v} = (v_j)$ and $\mathbf{t} = (t_j)$, $f(\mathbf{t}, \mathbf{v})$ denotes the vector with entries $f(t_j, v_j)$. If $b_0 = 0$ and $\mathbf{a} = 0$, then it follows from (1.3′) that (1.3) reduces to (1.2), so that in each step the computational complexity of (1.2) and (1.3) differ by the evaluation of $f(t_n, y_n)$, but

both methods require the solution of a system of dimension kd. Since the bulk of the computational effort goes into solving this system, the methods (1.2) and (1.3) may be considered as equally expensive.

The vectors $\mathbf{Y}_{n+1}$ and $\mathbf{c}$ will, respectively, be called the *stage vector* and the *block point vector*, and the points t_n and $t_n + c_j h$ will, respectively, be called *step points* and *block points*. The minimal order achieved at the block points and step points are, respectively, the *stage order* and *step point order*.

If the method parameters are chosen in such a way that the stage order is as large as possible with $\mathbf{c}$ arbitrary, then (1.3) is equivalent to the IRK method derived from Lagrange quadrature formulas and will be called a *Lagrange method*. If $c_j = j/k$, then Lagrange methods reduce to the *Newton-Cotes methods* studied in Watts and Shampine [23], and if the components of $\mathbf{c}$ equal the Lobatto quadrature points, then they reduce to the Lobatto IIIA methods. However, Newton-Cotes and Lobatto IIIA methods are only *weakly* A-stable (i.e., the method hardly damps the highly stiff components in the numerical error). It is our aim to construct Lagrange methods with better stability properties than Newton-Cotes and Lobatto IIIA methods, i.e., methods which damp both nonstiff and stiff components occurring in the numerical error (*strongly* A-stable methods).

An important family of IRK methods are the so-called *stiffly accurate methods* (cf. Alexander [1]). If the IRK method is of the form (1.3), then this family is obtained by setting

$$b_0 = \mathbf{e}_k^T \mathbf{a}, \qquad \mathbf{b}^T = \mathbf{e}_k^T A, \qquad c_k = 1, \tag{1.4}$$

where $\mathbf{e}_k$ is the kth unit vector. Notice that, when represented by their Butcher array (1.3), the last row in (1.3) equals the preceding one. It was shown by Hairer, Lubich, and Roche [11] that this property implies that for certain classes of stiff problems the method does not suffer the effect of order reduction. Examples of stiffly accurate IRK methods are the Lobatto IIIA, Radau IIA and Newton-Cotes methods.

1.3. Diagonally implicit iteration of IRK methods. After a finite number of m iterations of the implicit relation for $\mathbf{Y}_{n+1}$ given in (1.3′) by the aforementioned diagonally implicit iteration process (or briefly *diagonal iteration*) (see also § 3), the resulting scheme actually is an $(mk+1)$-stage DIRK method. One of these stages is explicit and the other mk stages are of diagonally implicit form. However, a large number of these mk implicit stages can be computed in parallel, resulting in a process where only m stages have to be computed sequentially.

The iteration parameters of the method can be tuned in such a way that we get fast convergence to the stability characteristics of the corrector method, provided that the corrector is stiffly accurate (in § 3.3.1, we will show that the diagonal iteration of the type employed in this paper is not suitable for iterating *nonstiffly accurate* correctors).

Second, it has been demonstrated that the iterated methods based on strongly A-stable correctors (such as the Radau IIA correctors and the Lagrange correctors derived in § 4) are within a few iterations strongly A-stable themselves. It is highly unlikely that this nice property is shared by the methods based on (weakly) A-stable IRK correctors because the stability function of the iterated methods should converge to a (weakly) A-acceptable function. In fact, for a number of Newton-Cotes and Lobatto IIIA correctors it was checked that the stability function becomes A-acceptable only after an infinite number of iterations.

Finally, numerical experiments reveal that the drop in accuracy, exhibited in many stiff problems by the conventionally constructed DIRK methods, is not shown by the

DIRK methods constructed by the diagonal iteration process of this paper. In a forthcoming paper [4] it is intended to present a theoretical analysis of this phenomenon using the error analysis proposed in Burrage [3].

2. Accuracy and stability of the corrector. In the following two subsections, we discuss the stage order, step point order, and stability of the corrector equation (1.3′).

2.1. Stage order. Let $\mathbf{Y}(t_{n+1})$ denote the vector with components $y(t_n+c_ih)$ where y is the locally exact solution of (1.1) satisfying $y(t_n)=y_n$, then, following Butcher [7], (1.3′) is said to have stage order r if the residual left upon substitution of $\mathbf{Y}(t_{n+1})$ into the formula for $\mathbf{Y}_{n+1}$ is of order $r+1$ in h, i.e.,

$$\mathbf{Y}(t_{n+1})-hAf(\mathbf{e}t_n+\mathbf{c}h,\mathbf{Y}(t_{n+1}))-\mathbf{e}y_n-h\mathbf{a}f(t_n,y_n)=O(h^{r+1}). \tag{2.1}$$

The stage-order conditions for (1.3′) are straightforwardly derived (cf. [22]) and are given by

$$\mathbf{C}_j=\mathbf{0},\quad j=1,\cdots,r,\quad \mathbf{C}_1:=\mathbf{a}+A\mathbf{e}-\mathbf{c},\quad \mathbf{C}_j:=jA\mathbf{c}^{j-1}-\mathbf{c}^j,\quad j=2,3,\cdots, \tag{2.2}$$

where $\mathbf{c}^j$ denotes the vector with components $(c_i)^j$. Thus, to achieve stage order r for a given block point vector $\mathbf{c}$, we have to solve rk linear equations in k^2+k unknowns, so that the maximal stage order equals $k+1$. The corresponding methods will be called *Lagrange methods.*

2.2. Step point order. Consider the formula for y_{n+1} given in (1.3′):

$$y_{n+1}=y_n+hb_0f(t_n,y_n)+h\mathbf{b}^Tf(\mathbf{e}t_n+\mathbf{c}h,\mathbf{Y}_{n+1}). \tag{2.3}$$

Since $\mathbf{Y}_{n+1}$ approximates $\mathbf{Y}(t_{n+1})$ with (local) order $r+1$, r being the stage order (cf. (2.1)), we can derive that y_{n+1} has (at least) order $p=\min\{r+1,q\}$ if the conditions

$$D_j=0,\quad j=1,\cdots,q,\quad D_1:=b_0+\mathbf{b}^T\mathbf{e}-1,\quad D_j:=j\mathbf{b}^T\mathbf{c}^{j-1}-1,\quad j=2,3,\cdots \tag{2.4}$$

are satisfied. We remark that p may be larger than $\min\{r+1,q\}$ if the methods possess the property of so-called "superconvergence" which for example is the case in Gauss, Radau, and Lobatto methods. The error constant of (2.3) is given by

$$E_{q+1}:=\frac{D_{q+1}}{(q+1)!}=\frac{(q+1)\mathbf{b}^T\mathbf{c}^q-1}{(q+1)!}. \tag{2.5}$$

Assuming that $\mathbf{c}$ is given, the conditions (2.4) present a linear system of q equations in $k+1$ unknowns, so that by setting $q=k+1$ we achieve at least step point order $p=\min\{r+1,k+1\}$ for any block point vector $\mathbf{c}$.

As already observed in the introduction, the usual approach in exploiting the vector $\mathbf{c}$ is the maximization of the step point order (to obtain "superconvergence"). Alternatively, we may use $\mathbf{c}$ for improving the stability of the method or for the minimization of error constants. In this paper, we shall use $\mathbf{c}$ for achieving strong A-stability.

In the special case of stiffly accurate methods satisfying condition (1.4), y_{n+1} equals the last component of $\mathbf{Y}_{n+1}$ so that the step point order p is also at least the stage order r, but is sometimes higher. For instance, the Newton–Cotes methods have stage order $k+1$ and, if k is even, step point order $k+2$.

2.3. Stability. By applying (1.3′) to the test equation $y'=\lambda y$, we are led to recursions of the form

$$\mathbf{Y}_{n+1}=[I-zA]^{-1}[\mathbf{e}+z\mathbf{a}]y_n, \quad y_{n+1}=(1+b_0z)y_n+z\mathbf{b}^T\mathbf{Y}_{n+1}, \quad z:=\lambda h. \tag{2.6}$$

Hence,

$$y_{n+1}=R(z)y_n, \qquad R(z):=1+b_0z+z\mathbf{b}^T[I-zA]^{-1}[\mathbf{e}+z\mathbf{a}]. \tag{2.7}$$

$R(z)$ is called the stability function of the one-step method. In the special case of stiffly accurate methods where (1.4) is satisfied, (2.7) reduces to

$$y_{n+1}=R(z)y_n, \qquad R(z):=\mathbf{e}_k^T[I-zA]^{-1}[\mathbf{e}+z\mathbf{a}]. \tag{2.8}$$

The stability region of the method is defined by the region where R is bounded by 1. In the case of the Newton-Cotes methods where the components of $\mathbf{c}$ are equally spaced, it was shown in Watts and Shampine [23] that they are A-stable for $k\leqq 8$ (but they are not for $k=9$ and $k=10$).

We conclude this section by summarizing in Table 2.1 the characteristics of a number of correctors available in the literature. In this table, it is assumed that the IRK method is presented in the form (1.3′), so that for all methods listed the dimension of the implicit relation to be solved equals kd, d being the dimension of the system of ODEs.

3. Diagonal iteration. We shall use a diagonal iteration method to solve the stage vector $\mathbf{Y}_{n+1}$ from the fully implicit (corrector) equation defined in (1.3′). For scalar differential equations, the iteration method reads

$$\begin{aligned}
&\mathbf{Y}^{(1)}-hDf(\mathbf{e}t_n+\mathbf{c}h,\mathbf{Y}^{(1)})=y_n\mathbf{e}+h\mathbf{a}f(t_n,y_n)+h[A-D]f(\mathbf{t}^{(0)},\mathbf{Y}^{(0)}),\\
&\mathbf{Y}^{(j)}-hDf(\mathbf{e}t_n+\mathbf{c}h,\mathbf{Y}^{(j)})=y_n\mathbf{e}+h\mathbf{a}f(t_n,y_n)\\
&\qquad\qquad +h[A-D]f(\mathbf{e}t_n+\mathbf{c}h,\mathbf{Y}^{(j-1)}), \qquad j=2,3,\cdots,
\end{aligned} \tag{3.1a}$$

where $(\mathbf{t}^{(0)},\mathbf{Y}^{(0)})$ is an initial approximation to $(\mathbf{e}t_n+\mathbf{c}h,\mathbf{Y}_{n+1})$ and D is an arbitrary diagonal matrix. If m iterations are performed, then y_{n+1} is defined by

$$y_{n+1}=y_n+hb_0f(t_n,y_n)+h\mathbf{b}^Tf(\mathbf{e}t_n+\mathbf{c}h,\mathbf{Y}^{(m)}) \quad \text{or} \quad y_{n+1}=\mathbf{e}_k^T\mathbf{Y}^{(m)}, \tag{3.1b}$$

respectively, for nonstiffly and stiffly accurate correctors (cf. (1.4)).

By virtue of the diagonal structure of D, the iterated method (3.1) is suitable for use on parallel processors because in each iteration the components of $\mathbf{Y}^{(j)}$ can be computed in parallel.

TABLE 2.1

Summary of characteristics of IRK *methods.*

Method	Stages	Order p	Stage order r	Stability	Stiffly accurate	Reference
Gauss-Legendre	k	$2k$	k	A-stable for all k	no	Butcher [7]
Lobatto IIIA	$k+1$	$2k$	$k+1$	A-stable for all k	yes	Dekker and Verwer [9]
Radau IIA	k	$2k-1$	k	L-stable for all k	yes	Butcher [7]
Newton-Cotes	$k+1$	$2[(k+2)/2]$	$k+1$	A-stable for $k\leqq 8$	yes	Watts and Shampine [23]
Lagrange	$k+1$	$k+1$	$k+1$	Strongly A-stable	yes	For $k\leqq 4$ see § 4

There are many possibilities for choosing the matrix D which we summarize below:

(i) $D = O$: this is the most simple choice and yields an *explicit* iteration method (fixed point or functional iteration). This approach was followed in Nørsett and Simonsen [20], Lie [18], van der Houwen and Sommeijer [12], and Burrage [3]. These papers deal with the iteration of implicit methods for solving *nonstiff* ODEs. In the case of *stiff* ODEs, we should use matrices $D \neq O$.

(ii) D is such that for a *prescribed* number of iterations the method has favourable stability characteristics like A-stability or L-stability. This approach was followed in van der Houwen, Sommeijer, and Couzy [13], where the corrector only serves for providing its order of accuracy. In fact, it was shown that one may even use *explicit* correctors and still can obtain A- and L-stability after the particular number of iterations and a suitable choice of the matrix D.

(iii) $D = \operatorname{diag}(A\mathbf{e})$ or $D = \operatorname{diag}(A)$: this choice leads to nonlinear Jacobi-type iteration. The few experiments we performed revealed that the convergence is rather poor, so that we dropped this option.

(iv) D is such that the *nonstiff* components in the iteration error are strongly damped. This type of diagonal iteration will be called *nonstiff iteration*. Nonstiff iteration can be achieved by minimizing the spectral radius of the matrix $A - D$ (see § 3.2). A large number of experiments showed that this is not the way to proceed, at least not in the case of the one-step initial approximations to $\mathbf{Y}_{n+1}$ used in this paper.

(v) D is such that the stability function $R_m(z)$ of the iterated method rapidly converges to the stability function $R_{\text{corr}}(z)$ of the corrector. Hence, the corrector not only serves for providing its order of accuracy as in [13], but the iterated method also reflects the (assumed) nice stability properties of the corrector. Within this "stability function approach" there are various approaches:

- $D^{-1}\mathbf{c} = A^{-1}\mathbf{c}$: this relation uniquely defines D provided that A is nonsingular. As observed by Hundsdorfer [15], such matrices D imply that the stability functions of the corrector and of the iterated method are identical at infinity. Although a few first experiments did not yet show satisfactory results, this option should be investigated more closely (see [4]).
- Minimization of the spectral radius of the matrix $I - D^{-1}A$. This choice implies that R_m converges fast to R_{corr} at infinity, but, at the same time, it also strongly damps the stiff components of the iteration error. This type of diagonal iteration will be called *stiff iteration*. It is the approach adopted in the present paper (see § 3.3). Our experiments in § 5 reveal that stiff iteration is suited for suppressing the phenomenon of order reduction within a few iterations, and in this respect, the methods of this paper perform much better than the methods proposed in [13].
- Other options as suggested by one of the referees, where some norm of $I - D^{-1}A$ is minimized rather than the spectral radius, or where $R_m - R_{\text{corr}}$ is minimized along the negative z-axis (or larger portions of the left halfplane), has not yet been tested and may turn out to be still more effective.

(vi) D is such that the lower order error terms in the truncation error are minimized. Since after a finite number of iterations the iterated method (3.1) formally is still a DIRK method and therefore suffers from order reduction, such an approach directly attacks the source for order reduction. This topic will also be considered in [4].

The approach of stiff iteration followed in this paper seems to be rather effective. However, by no means we do claim that this is the best way to proceed. In [4] we shall present more firm theoretical and experimental evidence of the merits of the various approaches for choosing the matrix D.

3.1. Computational costs. Each step of the (outer) iteration method (3.1a) requires the solution of a diagonally implicit relation. In order to solve this relation, we apply Newton iteration (inner iteration). There are various possibilities for starting the iteration method (3.1a) and the Newton iteration method, and for choosing the Jacobian matrix $J := \partial f/\partial y$ needed in the Newton iteration process. Obvious choices are listed in Table 3.1.

All possible combinations are equally expensive because the values of $f(t_n, y_n)$, $f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}^{(j-1)})$ and diag (J) are anyhow needed. The first-order approximations will reduce the magnitude of the smooth error components (low frequencies) more than the zero-order approximations do, but, unlike the zero-order approximations, they will also introduce stiff error components in the case of stiff differential equations. This particularly applies to the Jacobian matrix and the initial inner iterate because these approximations are needed in each outer iteration. Therefore, we shall only consider zero-order approximations to the Jacobian matrix and to the initial inner iterate (notice that in the case of systems of equations, the matrix J becomes a block-diagonal matrix). Furthermore, our experiments revealed that using zero-order approximations for the initial outer iterate is more robust than the above first-order approximations, and yields comparable accuracies. However, it should be observed that the topic of choosing suitable initial approximations to the stage vector (including multistep approximations in order to reduce the number of iterations) is extremely important and needs further research. Burrage [3] discussed this topic in the case of a general class of *explicit* predictor–corrector methods for nonstiff problems. His approach may be used to study initial approximations in the case of diagonally implicit predictor–corrector methods for stiff problems.

By performing m iterations, the method (3.1) may be considered as a DIRK method with $mk+1$ stages, of which one stage is explicit and the other mk stages are diagonally implicit. In fact, we may represent the method by the Butcher array:

$$
(3.1')\qquad
\begin{array}{c|cccccccl}
j=0 & 0 & & & & & & & \\
j=1 & \mathbf{c}-D\mathbf{e} & D & & & & & & \\
j=2 & \mathbf{a} & A-D & D & & & & & \\
j=3 & \mathbf{a} & O & A-D & D & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
j=m & \mathbf{a} & O & & \cdots & & O & A-D & D \\
\hline
 & b_0 & \mathbf{0}^T & & \cdots & & \mathbf{0}^T & \mathbf{0}^T & \mathbf{b}^T & \text{(nonstiffly accurate correctors)} \\
 & \mathbf{e}_k^T\mathbf{a} & \mathbf{0}^T & & \cdots & & \mathbf{0}^T & \mathbf{e}_k^T(A-D) & \mathbf{e}_k^T D & \text{(stiffly accurate correctors)}
\end{array}
.
$$

Since each iteration step in (3.1a) essentially requires the "wall clock time" involved in evaluating one component of $f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}^{(j-1)})$ and solving one system of dimension d, we conclude that, effectively, the work involved in performing one step by the DIRK

TABLE 3.1

Starting the inner and outer iteration processes.

Order of approximation	0	1
Jacobian matrix	diag $[J(\mathbf{e}t_n, \mathbf{e}y_n)]$	diag $[J(\mathbf{e}t_n + \mathbf{c}h, y_n\mathbf{e} + h\mathbf{c}f(t_n, y_n))]$
Initial iterate in (3.1a)	$\mathbf{Y}^{(0)} = y_n\mathbf{e}$, $\mathbf{t}^{(0)} = \mathbf{e}t_n$	$\mathbf{Y}^{(0)} = y_n\mathbf{e} + h\mathbf{c}f(t_n, y_n)$, $\mathbf{t}^{(0)} = \mathbf{e}t_n + \mathbf{c}h$
Initial Newton iterate	$\mathbf{Y}^{(j-1)}$	$y_n\mathbf{e} + h\mathbf{a}f(t_n, y_n) + hAf(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}^{(j-1)})$

method (3.1′) consists of

$$\begin{aligned}&(\text{evaluation of } f \text{ and } J)+(\text{LU decomposition of } I-d_jhJ)\\&\quad +m[\text{evaluation of } f+N(\text{forward/backward substitution}+\text{evaluation of } f)].\end{aligned} \tag{3.2}$$

In this expression N is defined by

$$N:=\frac{N_1+N_2+\cdots+N_m}{m}, \tag{3.3}$$

with N_j denoting the number of Newton iterations for computing that component of $\mathbf{Y}^{(j)}$ which requires the largest number of Newton iterations. Usually, the m iterations are the most expensive part of the total effort per step, and therefore we shall say that a DIRK method has m *effective* or *sequential* stages if there are m diagonally implicit systems to be solved.

3.1.1. Comparison with conventional DIRK methods. In the experiments reported in this paper, we used the stopping criterion that the Newton correction should be about the machine precision which is for our computer 10^{-14}. It turned out that N_j rapidly decreases with j which can be explained by observing that the initial iterate for starting the next inner iteration becomes more accurate when j increases. This is an advantage when compared with conventionally constructed DIRK methods already available in the literature (such DIRK methods will be indicated by "conventional" DIRK methods), because, for conventional DIRK methods, the number of Newton iterations for solving the implicit relations in the successive stages do, in general, not decrease.

In order to appreciate the computational costs of DIRK methods of type (3.1′), we should compare m with the number of sequential stages of conventional DIRK methods. In Table 3.2, the characteristics of such DIRK methods are listed together with the PARK and PDIRK methods derived in [16] and [13].

3.2. Order of accuracy. In order to analyse the order of accuracy of the iterated method (3.1), let $\mathbf{Y}(t_{n+1})$ denote the vector with components $y(t_n+c_ih)$ where y is the locally exact solution of (1.1). Then, in first approximation, we obtain

$$\begin{aligned}\mathbf{Y}(t_{n+1})-\mathbf{Y}^{(j)}&=[\mathbf{Y}(t_{n+1})-\mathbf{Y}_{n+1}]+[\mathbf{Y}_{n+1}-\mathbf{Y}^{(j)}]\\&=[\mathbf{Y}(t_{n+1})-\mathbf{Y}_{n+1}]+Z[\mathbf{Y}_{n+1}-\mathbf{Y}^{(j-1)}]\\&=[\mathbf{Y}(t_{n+1})-\mathbf{Y}_{n+1}]+Z^j[\mathbf{Y}_{n+1}-\mathbf{Y}^{(0)}], \qquad j=1,2,\cdots,\end{aligned} \tag{3.4a}$$

TABLE 3.2

Summary of characteristics of DIRK, PARK, *and* PDIRK *methods of order* $p\geqq 3$.

Order	Stage order	Sequential stages	Processors	Stability	Reference
$p=3$	1	$p-1$	1	A-stable	Nørsett [19]
$p=3$	2	$p-1$	1	Strongly A-stable	Crouzeix [8]
$p=4$	1	$p-1$	1	A-stable	Crouzeix [8], Alexander [1]
$p=4$	1	$p-2$	2	L-stable	Iserles and Nørsett [16]
$p=3,4,5$	1	$p-1$	$[(p+1)/2]$	Strongly A-stable	van der Houwen et al. [13]
$p=6,7$	1	$p-1$	$[(p+1)/2]$	Strongly $A(\alpha)$-stable	ibid
$p\leqq 6, p=8$	1	p	$[(p+1)/2]$	L-stable	ibid
$p=7,8,10$	1	$p+1$	$[(p+1)/2]$	L-stable	ibid

where Z is the iteration matrix defined by

$$Z = Z(hDJ) := [I - hDJ]^{-1}[AD^{-1} - I]hDJ, \tag{3.4b}$$

with J again denoting the Jacobian matrix of f.

Let r be the stage order of the corrector (1.3), then (cf. (2.1))

$$\mathbf{Y}(t_{n+1}) - \mathbf{Y}_{n+1} = O(h^{r+1}).$$

Since $Z = O(h)$ and $\mathbf{Y}_{n+1} - \mathbf{Y}^{(0)} = O(h)$, the local error of the stage vectors satisfy the order relation

$$\mathbf{Y}(t_{n+1}) - \mathbf{Y}^{(j)} = O(h^{r+1}) + O(h^{j+1}), \tag{3.5}$$

so that, after m iterations, (3.1) defines a method in which $\mathbf{Y}^{(m)}$ approximates $\mathbf{Y}(t_{n+1})$ with order $r^* = \min\{r, m\}$. We shall say that (3.1) has *stage order* r^* (although formally, when (3.1) is considered as a DIRK method, its stage order is only 1). Thus, the optimal stage-order methods, that is the methods based on the Lagrange methods as defined above, have stage order $r^* = k+1$ provided that at least $m = k+1$ iterations are performed.

In order to get more insight into the rate of convergence of the iteration process (3.1), we consider the test equation

$$\frac{dy(t)}{dt} = \lambda y(t), \tag{3.6}$$

where λ runs through the spectrum $\Lambda(J)$ of J. The matrix Z assumes the form

$$Z = zD[I - zD]^{-1}[D^{-1}A - I] = z[I - zD]^{-1}[A - D], \qquad z := \lambda h. \tag{3.7}$$

Suppose that J has a complete eigensystem, and let us call the eigenvectors of hJ corresponding to the eigenvalues of large and small modulus, respectively *stiff* and *nonstiff* components. From (3.7) we see that for the nonstiff components (i.e., corresponding to small values of $|z|$) the matrix Z behaves approximately as $z[A - D]$. Hence, these components in the iteration error are strongly damped if the matrix $A - D$ has eigenvalues of small magnitude. Thus, rapid convergence of the nonstiff components is obtained by minimizing the spectral radius of $A - D$. However, as already remarked above, such a *nonstiff* iteration process gives a poor overall convergence. Alternatively, for the stiff components (i.e., corresponding to large values of $|z|$), the matrix Z behaves as $-D^{-1}[A - D]$. Hence, a strong damping of these components requires the minimization of the spectral radius of $I - D^{-1}A$, leading to *stiff* iteration. In the following section, we shall see that this condition also plays a role in the stability of the iterated method.

3.3. Stability. One may argue that there is no reason to continue the iteration process after $m = r$ iterations, because the stage errors of the corrector and of the iterated method have become of the same order in h and may therefore be expected to be of comparable magnitude. However, there is no guarantee that after $m = r$ iterations the *stability properties* of (1.3′) are also comparable with those of the corrector. This brings us to consider the stability of the DIRK method (3.1′). In order to see how the stability depends on the number of iterations m, we apply the method to the test equation (3.6), so that (3.1a) reduces to

$$\mathbf{Y}^{(m)} = (Z^m \mathbf{e} + [I - Z]^{-1}[I - Z^m][I - zD]^{-1}(\mathbf{e} + z\mathbf{a}))y_n.$$

We shall discuss the stability of iterating a nonstiffly accurate and a stiffly accurate corrector separately.

3.3.1. Nonstiffly accurate correctors. If y_{n+1} is computed by means of the formula

$$y_{n+1} = [1 + zb_0]y_n + z\mathbf{b}^T\mathbf{Y}^{(m)},$$

then it can be expressed as

$$y_{n+1} = [1 + zb_0 + z\mathbf{b}^T(Z^m\mathbf{e} + [I - Z]^{-1}[I - Z^m][I - zD]^{-1}(\mathbf{e} + z\mathbf{a}))]y_n, \tag{3.8}$$

so that the stability function is given by

$$R_m(z) := 1 + zb_0 + z\mathbf{b}^T(Z^m\mathbf{e} + [I - Z]^{-1}[I - Z^m][I - zD]^{-1}(\mathbf{e} + z\mathbf{a})). \tag{3.9}$$

It is easily verified that this function can be written in the form

$$R_m(z) := 1 + zb_0 + z\mathbf{b}^T[I - zA]^{-1}(\mathbf{e} + z\mathbf{a}) - z^2\mathbf{b}^TZ^m[I - zA]^{-1}(A\mathbf{e} + \mathbf{a}).$$

Assuming that the stage order of the corrector is at least one, we may set $A\mathbf{e} + \mathbf{a} = \mathbf{c}$ (see (2.2)), so that

$$R_m(z) := R_{\text{corr}}(z) - z^2\mathbf{b}^T[Z(zD)]^m[I - zA]^{-1}\mathbf{c}, \tag{3.10}$$

where R_{corr} denotes the stability function of the corrector given by (2.7). Finally, on substitution of (3.7) into (3.10) we obtain

$$R_m(z) = R_{\text{corr}}(z) - z^{m+2}\mathbf{b}^T([I - zD]^{-1}[A - D])^m[I - zA]^{-1}\mathbf{c}. \tag{3.11}$$

From this expression we can derive the convergence behaviour of R_m to R_{corr} for large values of $|z|$:

$$R_m(z) = R_{\text{corr}}(z) + z\mathbf{b}^T[I - D^{-1}A]^mA^{-1}\mathbf{c} \quad \text{as } |z| \to \infty,$$

showing that for any fixed m the stability function becomes unbounded as $|z|$ tends to infinity, unless the matrix D is such that

$$\mathbf{b}^T[I - D^{-1}A]^mA^{-1}\mathbf{c} = 0.$$

Writing this equation as

$$\mathbf{b}^T[I - D^{-1}A]^{m-1}[I - D^{-1}A]A^{-1}\mathbf{c} = \mathbf{b}^T[I - D^{-1}A]^{m-1}[A^{-1}\mathbf{c} - D^{-1}\mathbf{c}] = 0,$$

we see that it can be satisfied for all m if we choose D such that [15]

$$D^{-1}\mathbf{c} = A^{-1}\mathbf{c}. \tag{3.12}$$

Unfortunately, a few first experiments showed that the performance of the corresponding method (3.1′) is not satisfactory (see § 5.3). Therefore, we conclude that diagonal iteration as defined by (3.1′) is in general not suitable for iterating nonstiffly accurate correctors and excludes the Gauss-Legendre formulas as suitable corrector methods. However, it should be remarked that by defining the initial iterate $\mathbf{Y}^{(0)}$ implicitly, rather than just setting $\mathbf{Y}^{(0)} = y_n\mathbf{e}$, the above stability problem can be avoided (cf. [13]), so that the matrix D remains available for improving the performance of the iteration process. As observed in § 3.1, the topic of finding suitable initial approximations to the stage vector in diagonally iterated RK methods deserves further research, but will not be an issue in this paper.

3.3.2. Stiffly accurate correctors. In the stiffly accurate case where y_{n+1} is computed by means of the formula

$$y_{n+1} = \mathbf{e}_k^T\mathbf{Y}^{(m)},$$

we arrive at the stability function

$$R_m(z) = R_{\text{corr}}(z) - z\mathbf{e}_k^T[Z(zD)]^m[I - zA]^{-1}\mathbf{c}, \tag{3.13}$$

where R_{corr} is defined by (2.8). We may express this function in the form

$$R_m(z) = R_{\text{corr}}(z) - [\sigma_m(z)]^m, \tag{3.13'}$$

where

$$\begin{aligned}\sigma_m(z) &:= [z\mathbf{e}_k^T[Z(zD)]^m[I - zA]^{-1}\mathbf{c}]^{1/m} \\ &= [z^{m+1}\mathbf{e}_k^T([I - zD]^{-1}[A - D])^m[I - zA]^{-1}\mathbf{c}]^{1/m}.\end{aligned}$$

For fixed values of m and assuming that D has positive diagonal elements, the function $\sigma_m(z)$ is bounded for all z in the closed left halfplane. This suggests to characterize the rate of convergence of R_m to R_{corr} by means of $\sigma_m(z)$. We shall call $\sigma_m(z)$ the *convergence factor associated with z*. For example, we have

$$\sigma_m(0) = 0, \quad \sigma_m(\infty) := [-\mathbf{e}_k^T[Z(-\infty)]^m A^{-1}\mathbf{c}]^{1/m} = [-\mathbf{e}_k^T[I - D^{-1}A]^m A^{-1}\mathbf{c}]^{1/m}. \tag{3.14}$$

Ideally, in order to get fast convergence of the stability function $R_m(z)$ to that of the corrector, we should try to minimize $\sigma_m(z)$ in the closed left halfplane. However, since in actual computation m is determined by some error criterion, we do not know m in advance, so that such an approach may be unattractive, particularly for larger values of k where more values of m have to be considered. Nevertheless, in a future paper [4], this possibility will be studied more closely in order to get further insight into how crucial the choice of D really is.

Another possibility is the minimization of $\sigma_m(z)$ for the highly stiff components (large values of $|z|$), because (3.14) shows that $\sigma_m(z)$ is already small for the nonstiff components. The most simple way to achieve this determines D according to (3.12), so that $\sigma_m(\infty)$ vanishes for all m [15]. In the experiments done so far, the convergence of the corresponding iteration process (3.1) is not satisfactory.

However, by choosing the matrix D, for a given corrector, such that the spectral radius of $Z(-\infty) = I - D^{-1}A$ is minimized over all possible diagonal matrices D with positive entries, we obtained a satisfactory convergence behaviour in a large number of experiments (see § 5, and the Appendix to [14]). The better convergence may be explained by observing that in this way, not only the value of $\sigma_m(\infty)$ is expected to be small (cf. (3.14)), but as already shown in § 3.2, at the same time the stiff components in the iteration error are strongly damped.

Together with the computation of the matrix D (cf. § 4), we computed, as a posteriori test, for a few values of m the "worst" convergence factor defined by

$$\sigma_m := \max_{\operatorname{Re} z \le 0} |\sigma_m(z)|. \tag{3.15}$$

Because $\sigma_m(z)$ is an analytic function in the closed left halfplane, its maximum is assumed on the boundary, i.e., on the imaginary axis.

In calculating σ_m it turned out that this quantity is larger than 1 for small values of m but rather quickly decreases to a moderate size as m increases. The values of σ_m show by what factor the (maximal) difference between the two stability functions is reduced in each iteration if we continue to iterate when the stage order of the corrector has been reached. Due to the fact that $\sigma_m > 1$ for small m, it is likely that the corresponding iterated method is not A-stable. On the other hand, assuming that the iteration process (3.1) is convergent, we know that $[\sigma_m(z)]^m \to 0$ for $m \to \infty$, i.e., $R_m(z)$ converges to the A-acceptable stability function $R_{\text{corr}}(z)$. Therefore, it is of interest to know the minimal value of m such that $R_m(z)$ is A-acceptable for all m equal to or larger than this minimal value. This for the iteration process critical number of iterations will be denoted by m_{crit}. Evidently, the value of m_{crit} is expected to be large if the

corrector is not strongly A-stable. In order to illustrate this, we considered the methods using weakly A-stable Newton–Cotes and Lobatto IIIA correctors (cf. Table 2.1) with minimized spectral radius of $I-D^{-1}A$. We verified that (for z in the closed left halfplane) the value of $\max|R_m(z)|\downarrow 1$ as $m\to\infty$, so that A-stability is only obtained in the limit. Hence, the Lobatto IIIA and the Newton–Cotes formulas seem to be less suitable as corrector methods. For the strongly A-stable Lagrange correctors and the L-stable Radau IIA correctors however, we found modest values of m_{crit}, so that after a few iterations the resulting method is already A-stable (see § 4).

4. Construction of methods. In this section, we consider a number of stiffly accurate correctors and we will construct the corresponding matrices D for use on two-, three- or four-processor computers (i.e., methods of dimension $k=2,3,4$).

For $k=2$, we shall give a rather detailed derivation, because in this case, it is still possible to construct suitable matrices D analytically. We derive matrices D for correctors of Newton–Cotes, Lobatto IIIA, strongly A-stable Lagrange, Radau IIA, and Gauss–Legendre type. The Gauss–Legendre method is not stiffly accurate, and therefore not suitable for diagonal iteration of type (3.1′), but it is included to demonstrate its unstable performance. For $k>2$, we resort to numerical search methods for finding suitable matrices D. Here, we refrained from looking for D matrices for the Gauss–Legendre method because of the rather poor two-processor results. In § 4.4 a summary of the main properties of the various methods is given.

It may be of interest to note that in our numerical search for strongly A-stable correctors we encountered strong numerical evidence for the following conjecture.

Conjecture. A necessary condition for a stiffly accurate Lagrange method as defined in § 1.2 to be strongly A-stable is

$$\sum_{j=1}^{k} c_j > \frac{k+1}{2}. \qquad \square$$

In order to save space, the correctors are presented by means of the matrix A and the vectors $\mathbf{a}$ and $\mathbf{c}$, and the iterated versions by only giving the matrix D, because, together with the corrector, D completely defines the iterated method. In the following, we only consider *stiff* iteration, that is, the construction of D will always be based on the minimization of the spectral radius $\rho(I-D^{-1}A)$ of the matrix $I-D^{-1}A$. If the entries of D are not exact (i.e., for $k\geqq 3$), then they are approximated by rational expressions. In addition to D, we present the values of $\rho(I-D^{-1}A)$, the range for σ_m with $r\leqq m\leqq 10$, the corresponding interval I_σ on the imaginary axis where the maxima are assumed, and the value of m_{crit} are given (cf. § 3.3.2). Finally, the stage and step point orders of the method are denoted by r and p, respectively.

4.1. Two-processor methods.

4.1.1. Lagrange methods. Let us first consider two-dimensional Lagrange methods ($k=2$) satisfying the condition (1.4). The stage-order conditions (2.2) can be solved for $r=3$ and yield the stiffly accurate Lagrange method

$$A=\frac{1}{6(1-c)}\begin{pmatrix} c(3-2c) & -c^3 \\ c^{-1} & 2-3c \end{pmatrix}, \qquad \mathbf{a}=\frac{1}{6(1-c)}\begin{pmatrix} 3c-4c^2+c^3 \\ -c^{-1}+4-3c \end{pmatrix},$$
$$\mathbf{c}=\begin{pmatrix} c \\ 1 \end{pmatrix}, \qquad p=r=3, \tag{4.1a}$$

where c is a free parameter (recall that $p=4$ if $c=\frac{1}{2}$). An elementary calculation shows

that the stability function of (4.1a) is given by

$$R(z)=\frac{6+2(2-c)z+(1-c)z^2}{6-2(c+1)z+cz^2}. \tag{4.2}$$

This function is A-acceptable for $c\geqq\frac{1}{2}$ and strongly A-acceptable for $c>\frac{1}{2}$.

Next, we determine the matrix D in (3.1). It is convenient to write

$$D=\frac{1}{6(1-c)}\begin{pmatrix}1/\delta_1 & 0\\ 0 & 1/\delta_2\end{pmatrix},$$

so that

$$I-D^{-1}A=\begin{pmatrix}1-c(3-2c)\delta_1 & c^3\delta_2\\ -c^{-1}\delta_1 & 1-(2-3c)\delta_2\end{pmatrix}.$$

The eigenvalues of $I-D^{-1}A$ satisfy the equation

$$\mu^2-S\mu+P=0,\qquad S:=2-c(3-2c)\delta_1-(2-3c)\delta_2,$$
$$P:=[1-c(3-2c)\delta_1][1-(2-3c)\delta_2]+c^2\delta_1\delta_2.$$

By setting $S=P=0$ we achieve that $\rho(I-D^{-1}A)$ vanishes. The parameters δ_1 and δ_2 then satisfy the equations

$$c(3-2c)\delta_1+(2-3c)\delta_2=2,\qquad [1-c(3-2c)\delta_1]^2-c^2\delta_1\delta_2=0,$$

leading to

$$\delta_1=\frac{1+Q}{c(3-2c)},\qquad \delta_2=\frac{1-Q}{2-3c},\qquad Q:=\frac{\pm\sqrt{6c}}{6(1-c)},$$

so that the matrix D is given by

$$D=\frac{1}{6(1-c)}\begin{pmatrix}c(3-2c)/(1+Q) & 0\\ 0 & (2-3c)/(1-Q)\end{pmatrix},\quad \rho(I-D^{-1}A)=0. \tag{4.1b}$$

The iterated Lagrange method with zero convergence factor at infinity is completely determined by the corrector (4.1a) and the matrix (4.1b).

For $c=\frac{1}{2}$ we derive from (4.1a) the Newton-Cotes corrector (with $p=4$ and $r=3$)

$$A=\frac{1}{24}\begin{pmatrix}8 & -1\\ 16 & 4\end{pmatrix},\qquad \mathbf{a}=\frac{1}{24}\begin{pmatrix}5\\ 4\end{pmatrix},\qquad \mathbf{c}=\begin{pmatrix}\frac{1}{2}\\ 1\end{pmatrix}. \tag{4.3a}$$

We observe that this corrector coincides with the three-stage Lobatto IIIA method. The stability function R of (4.3a) reduces to the $(2,2)$ Padé approximation to the exponential function. Recall that R is A-acceptable but not *strongly* A-acceptable. From (4.1b) we obtain the matrix

$$\begin{gathered}D=\frac{1}{2}\begin{pmatrix}2/(3+\sqrt{3}) & 0\\ 0 & 1/(3-\sqrt{3})\end{pmatrix},\qquad \rho(I-D^{-1}A)=0,\\ \sigma_m\in[0.21,0.36],\qquad I_\sigma=[3.9i,5.1i],\qquad m_{\text{crit}}=\infty.\end{gathered} \tag{4.3b}$$

A natural question now is, whether it is possible to choose c such that the stability is improved. Unfortunately, (4.1a) shows that it is not possible to achieve L-stability

(which would require $c=1$), but strong A-stability is obtained for $c>\frac{1}{2}$. For example, by choosing $c=\frac{3}{4}$ we have $R(\infty)=\frac{1}{3}$. The corresponding Lagrange method is defined by

$$A=\frac{1}{288}\begin{pmatrix}216 & -81\\ 256 & -48\end{pmatrix},\qquad \mathbf{a}=\frac{1}{288}\begin{pmatrix}81\\ 80\end{pmatrix},\qquad \mathbf{c}=\begin{pmatrix}\frac{3}{4}\\ 1\end{pmatrix} \tag{4.4a}$$

for which $p=r=3$. The iterated version is defined by

$$\begin{gathered} D=\frac{1}{12}\begin{pmatrix}9/(\sqrt{2}+1) & 0\\ 0 & 2/(\sqrt{2}-1)\end{pmatrix},\qquad \rho(I-D^{-1}A)=0,\\ \sigma_m\in[0.21,0.33],\qquad I_\sigma=[3.2i,4.1i],\qquad m_{\text{crit}}=2. \end{gathered} \tag{4.4b}$$

4.1.2. Gauss and Radau methods. As reference methods for our numerical experiments, we take the conventional two-stage Gauss–Legendre and Radau IIA methods. The Gauss–Legendre corrector, and its iterated version is defined by

$$\begin{gathered} A=\frac{1}{12}\begin{pmatrix}3 & 3-2\sqrt{3}\\ 3+2\sqrt{3} & 3\end{pmatrix},\qquad \mathbf{a}=\mathbf{0},\qquad b_0=0,\\ \mathbf{b}=\frac{1}{2}\mathbf{e},\qquad \mathbf{c}=\frac{1}{12}\begin{pmatrix}6-2\sqrt{3}\\ 6+2\sqrt{3}\end{pmatrix},\qquad p=4,\qquad r=2, \end{gathered} \tag{4.5a}$$

$$D=\frac{1}{6}\begin{pmatrix}1 & 0\\ 0 & 3\end{pmatrix},\qquad \rho(I-D^{-1}A)=0. \tag{4.5b}$$

The Radau IIA-based method is given by

$$\begin{gathered} A=\frac{1}{12}\begin{pmatrix}5 & -1\\ 9 & 3\end{pmatrix},\qquad \mathbf{a}=\mathbf{0},\qquad b_0=0,\qquad \mathbf{b}^T=\mathbf{e}_2^TA,\\ \mathbf{c}=\begin{pmatrix}\frac{1}{3}\\ 1\end{pmatrix},\qquad p=3,\qquad r=2, \end{gathered} \tag{4.6a}$$

$$\begin{gathered} D=\frac{1}{30}\begin{pmatrix}20-5\sqrt{6} & 0\\ 0 & 12+3\sqrt{6}\end{pmatrix},\qquad \rho(I-D^{-1}A)=0,\\ \sigma_m\in[0.27,0.35],\qquad I_\sigma=[2.6i,3.7i],\qquad m_{\text{crit}}=1. \end{gathered} \tag{4.6b}$$

4.2. Three-processor methods.

4.2.1. Newton–Cotes method. For $k=3$ and equidistant abscissas the corrector is given by

$$\begin{gathered} A=\frac{1}{72}\begin{pmatrix}19 & -5 & 1\\ 32 & 8 & 0\\ 27 & 27 & 9\end{pmatrix},\qquad \mathbf{a}=\frac{1}{72}\begin{pmatrix}9\\ 8\\ 9\end{pmatrix},\\ b_0=\mathbf{e}_3^T\mathbf{a},\qquad \mathbf{b}^T=\mathbf{e}_3^TA,\qquad \mathbf{c}=\frac{1}{3}\begin{pmatrix}1\\ 2\\ 3\end{pmatrix} \end{gathered} \tag{4.7a}$$

with $p=r=4$, and with A-acceptable stability function (see Watts and Shampine [23]
By a numerical search we found the matrix

$$D=\begin{pmatrix} \frac{897}{7303} & 0 & 0 \\ 0 & \frac{2485}{10968} & 0 \\ 0 & 0 & \frac{8980}{27627} \end{pmatrix}, \qquad \rho(I-D^{-1}A)\approx 0.01,$$

(4.7b)

$$\sigma_m \in [0.49, 0.77], \qquad I_\sigma=[7.1i, 8.4i], \qquad m_{\text{crit}}=\infty.$$

4.2.2. Lobatto IIIA method. For $k=3$ and Lobatto abscissas the corrector is giv
by

$$A=\frac{1}{120}\begin{pmatrix} 25-\sqrt{5} & 25-13\sqrt{5} & -1+\sqrt{5} \\ 25+13\sqrt{5} & 25+\sqrt{5} & -1-\sqrt{5} \\ 50 & 50 & 10 \end{pmatrix}, \qquad \mathbf{a}=\frac{1}{120}\begin{pmatrix} 11+\sqrt{5} \\ 11-\sqrt{5} \\ 10 \end{pmatrix},$$

(4.8a)

$$b_0=\mathbf{e}_3^T\mathbf{a}, \qquad \mathbf{b}^T=\mathbf{e}_3^T A, \qquad \mathbf{c}=\frac{1}{10}\begin{pmatrix} 5-\sqrt{5} \\ 5+\sqrt{5} \\ 10 \end{pmatrix},$$

with $p=6$ and $r=4$, and with A-acceptable stability function (see Dekker and Verv
[9]). The iterated version is generated by

$$D=\begin{pmatrix} \frac{2661}{5542} & 0 & 0 \\ 0 & \frac{754}{6891} & 0 \\ 0 & 0 & \frac{1567}{9771} \end{pmatrix}, \qquad \rho(I-D^{-1}A)\approx 0.0043,$$

(4.8b)

$$\sigma_m \in [0.52, 0.88], \qquad I_\sigma=[8.9i, 10i], \qquad m_{\text{crit}}=\infty.$$

4.2.3. Lagrange method. By keeping c_1 and c_2 free, we can construct *stro*
A-stable methods with stage order four. It can be shown that the stability functio
A-acceptable for $c_1+c_2=1$ and strongly A-acceptable for $c_1+c_2>1$. A numerical sea
produced the block point vector $\mathbf{c}=(\frac{7}{12}, \frac{5}{6}, 1)^T$ for which parameter values of accept
magnitude and a damping factor $|R(\infty)|\approx 0.143$ are obtained. The correspon
corrector reads

$$A=\frac{1}{120960}\begin{pmatrix} 98392 & -81634 & 31213 \\ 112000 & -61600 & 28000 \\ 110592 & -48384 & 36288 \end{pmatrix}, \qquad \mathbf{a}=\frac{1}{120960}\begin{pmatrix} 22589 \\ 22400 \\ 22464 \end{pmatrix},$$

(4.9a)

$$b_0=\mathbf{e}_3^T\mathbf{a}, \qquad \mathbf{b}^T=\mathbf{e}_3^T A, \qquad \mathbf{c}=\frac{1}{12}\begin{pmatrix} 7 \\ 10 \\ 12 \end{pmatrix}$$

with $p=r=4$. The iterated method is generated by

$$D=\begin{pmatrix} \frac{2246}{10669} & 0 & 0 \\ 0 & \frac{2537}{8794} & 0 \\ 0 & 0 & \frac{3026}{8923} \end{pmatrix}, \qquad \rho(I-D^{-1}A)\approx 0.011,$$

(4.9b)

$$\sigma_m \in [0.49, 0.69], \qquad I_\sigma=[5.1i, 6.2i], \qquad m_{\text{crit}}=3.$$

4.2.4. Radau method. The 3-stage Radau IIA corrector is defined by [7]

$$A=\begin{pmatrix} \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{pmatrix}, \tag{4.10a}$$

$$\mathbf{a}=\mathbf{0}, \qquad \mathbf{b}_0=0, \qquad \mathbf{b}^T=\mathbf{e}_3^T A, \qquad \mathbf{c}=A\mathbf{e}$$

with $p=5$, $r=3$, and L-acceptable stability function. The matrix D is given by

$$D=\begin{pmatrix} \frac{4365}{13624} & 0 & 0 \\ 0 & \frac{1032}{7373} & 0 \\ 0 & 0 & \frac{1887}{5077} \end{pmatrix}, \qquad \rho(I-D^{-1}A)\approx 0.0047, \tag{4.10b}$$

$$\sigma_m\in[0.52,\,1.0], \qquad I_\sigma=[6.6i,\,9.3i], \qquad m_{\text{crit}}=5.$$

4.3. Four-processor methods.

4.3.1. Newton–Cotes method. For $k=4$ and equidistant abscissas the corrector is given by

$$A=\frac{1}{2880}\begin{pmatrix} 646 & -264 & 106 & -19 \\ 992 & 192 & 32 & -8 \\ 918 & 648 & 378 & -27 \\ 1024 & 384 & 1024 & 224 \end{pmatrix}, \qquad \mathbf{a}=\frac{1}{2880}\begin{pmatrix} 251 \\ 232 \\ 243 \\ 224 \end{pmatrix}, \tag{4.11a}$$

$$b_0=\mathbf{e}_4^T\mathbf{a}, \qquad \mathbf{b}^T=\mathbf{e}_4^T A, \qquad \mathbf{c}=\frac{1}{4}\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

with $p=6$, $r=5$, and with A-acceptable stability function. A numerical search did not produce a better matrix D than

$$D=\begin{pmatrix} \frac{992}{10759} & 0 & 0 & 0 \\ 0 & \frac{1365}{8107} & 0 & 0 \\ 0 & 0 & \frac{2709}{11281} & 0 \\ 0 & 0 & 0 & \frac{1717}{5549} \end{pmatrix}, \qquad \rho(I-D^{-1}A)\approx 0.1, \tag{4.11b}$$

$$\sigma_m\in[0.76,\,1.04], \qquad I_\sigma=[8.7i,\,11.8i], \qquad m_{\text{crit}}=\infty.$$

4.3.2. Lobatto IIA method. For $k=4$ and Lobatto abscissas the corrector is given by

$$A=\begin{pmatrix} \frac{343-9\sqrt{21}}{2520} & \frac{392-96\sqrt{21}}{2205} & \frac{343-69\sqrt{21}}{2520} & \frac{-21+3\sqrt{21}}{1960} \\ \frac{392+105\sqrt{21}}{2880} & \frac{8}{45} & \frac{392-105\sqrt{21}}{2880} & \frac{3}{320} \\ \frac{343+69\sqrt{21}}{2520} & \frac{392+96\sqrt{21}}{2205} & \frac{343+9\sqrt{21}}{2520} & \frac{-21-3\sqrt{21}}{1960} \\ \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \end{pmatrix}, \tag{4.12a}$$

$$\mathbf{a}=\begin{pmatrix}\frac{119+3\sqrt{21}}{1960}\\ \frac{13}{320}\\ \frac{119-3\sqrt{21}}{1960}\\ \frac{1}{20}\end{pmatrix},\qquad b_0=\mathbf{e}_4^T\mathbf{a},\qquad \mathbf{b}^T=\mathbf{e}_4^TA,\qquad \mathbf{c}=\begin{pmatrix}\frac{7-\sqrt{21}}{14}\\ \frac{1}{2}\\ \frac{7+\sqrt{21}}{14}\\ 1\end{pmatrix}$$

with $p=8$, $r=5$, and with A-acceptable stability function. A numerical search produced the matrix

$$D=\begin{pmatrix}\frac{2964}{9943} & 0 & 0 & 0\\ 0 & \frac{1875}{10334} & 0 & 0\\ 0 & 0 & \frac{608}{9403} & 0\\ 0 & 0 & 0 & \frac{3799}{23419}\end{pmatrix},\qquad \rho(I-D^{-1}A)\approx 0.021, \tag{4.12b}$$

$$\sigma_m\in[0.87,1.32],\qquad I_\sigma=[15.4i,19i],\qquad m_{\text{crit}}=\infty.$$

4.3.3. Lagrange method. Numerically, we found that the stability function is A-acceptable for $c_1+c_2+c_3=\frac{3}{2}$ and strongly A-acceptable for $c_1+c_2+c_3>\frac{3}{2}$. For $\mathbf{c}=(\frac{1}{6},\frac{7}{12},\frac{11}{12},1)^T$ we obtained parameter values of acceptable magnitude and a damping factor $|R(\infty)|\approx 0.325$. The corresponding corrector with $p=r=5$ reads

$$A=\frac{1}{49896000}\begin{pmatrix}5452832 & -872784 & 926800 & -556248\\ 17484082 & 13296591 & -6182575 & 3486252\\ 16192946 & 22005423 & 7263025 & -1229844\\ 16232832 & 21897216 & 9676800 & 598752\end{pmatrix},$$

$$\mathbf{a}=\frac{1}{332640}\begin{pmatrix}22436\\ 6811\\ 10043\\ 9936\end{pmatrix},\qquad b_0=\mathbf{e}_4^T\mathbf{a},\qquad \mathbf{b}^T=\mathbf{e}_4^TA,\qquad \mathbf{c}=\frac{1}{12}\begin{pmatrix}2\\ 7\\ 11\\ 12\end{pmatrix}. \tag{4.13a}$$

The iterated method is generated by

$$D=\begin{pmatrix}\frac{5147}{38467} & 0 & 0 & 0\\ 0 & \frac{1983}{17459} & 0 & 0\\ 0 & 0 & \frac{3197}{14090} & 0\\ 0 & 0 & 0 & \frac{3086}{12339}\end{pmatrix},\qquad \rho(I-D^{-1}A)\approx 0.045, \tag{4.13b}$$

$$\sigma_m\in[0.59,0.93],\qquad I_\sigma=[8.2i,11.8i],\qquad m_{\text{crit}}=6.$$

4.3.4. Radau method. The four-stage Radau IIA corrector reads

$$A=\begin{pmatrix}.11299947932316 & -.04030922072352 & .02580237742034 & -.0099046765073\\ .23438399574740 & .20689257393536 & -.04785712804854 & .01604742280652\\ .21668178462325 & .40612326386737 & .18903651817006 & -.02418210489983\\ .22046221117677 & .38819346884317 & .32884431998006 & \frac{1}{16}\end{pmatrix}, \tag{4.14a}$$

$$\mathbf{a}=\mathbf{0},\qquad b_0=0,\qquad \mathbf{b}^T=\mathbf{e}_4^TA,\qquad \mathbf{c}=A\mathbf{e}$$

with $p=7$, $r=4$, and with L-acceptable stability function. The iterated method is generated by

$$D=\begin{pmatrix} \frac{3055}{9532} & 0 & 0 & 0 \\ 0 & \frac{531}{5956} & 0 & 0 \\ 0 & 0 & \frac{1471}{8094} & 0 \\ 0 & 0 & 0 & \frac{1848}{7919} \end{pmatrix}, \qquad \rho(I-D^{-1}A)\approx 0.024, \tag{4.14b}$$

$$\sigma_m \in [0.74, 1.31], \qquad I_\sigma=[10.0i, 17.2i], \qquad m_{\text{crit}}=7.$$

4.4. Survey of methods. In Table 4.1, we have summarized a few characteristics of the methods derived in the preceding sections. In this table, the value of the step point order p corresponds to values of m equal to or greater than p, and the value of the stage order r corresponds to that of the corrector. From a computational point of view, the Lagrange and Radau IIA methods are the most attractive ones, because m_{crit} is relatively small. Thus, if these methods are implemented with some local error strategy for automatically estimating the number of iterations m and the stepsize h needed to meet local error tolerance, then the value of the "computational efficiency" quantity mL/h for integrating an interval of length L will not be unnecessarily large because of the development of instabilities. This observation is confirmed by the numerical experiments in § 5.4.

5. Numerical experiments. In this paragraph, the (stiff) diagonal iteration method developed above will be tested by integrating a number of stiff test problems. Section 5.1 presents these test problems. Section 5.2 compares the effective orders of Gauss-Legendre, Newton-Cotes, Lobatto IIIA, Radau IIA and Lagrange correctors, and in § 5.3, the performance of the diagonal iteration process with respect to the number of iterations is tested for a few two-processor correctors. Finally, in § 5.4, we compare the efficiency of the iterated methods with a few DIRK methods from the literature.

We recall that we only used the zero-order approximations to the Jacobian matrix and to the initial inner and outer iterates. In the tables of results, the accuracy of the

TABLE 4.1

Main characteristics of diagonally iterated IRK *methods.*

Method	p	r	k	$\rho(I-D^{-1}A)$	σ_m-range ($r \leqq m \leqq 10$)	m_{crit}
Newton-Cotes (4.3)	4	3	2	0	[0.21, 0.36]	∞
Lagrange (4.4)	3	3	2	0	[0.21, 0.33]	2
Radau IIA (4.6)	3	2	2	0	[0.27, 0.35]	1
Gauss (4.5)	4	2	2	0	∞	∞
Newton-Cotes (4.7)	4	4	3	0.008	[0.49, 0.77]	∞
Lobatto IIIA (4.8)	6	4	3	0.0043	[0.52, 0.88]	∞
Lagrange (4.9)	4	4	3	0.01	[0.49, 0.69]	3
Radau IIA (4.10)	5	3	3	0.0047	[0.52, 1.0]	5
Newton-Cotes (4.11)	6	5	4	0.1	[0.76, 1.04]	∞
Lobatto IIIA (4.12)	8	5	4	0.021	[0.87, 1.32]	∞
Lagrange (4.13)	5	5	4	0.045	[0.59, 0.93]	6
Radau IIA (4.14)	7	4	4	0.024	[0.74, 1.31]	7

results is given by means of the number of correct digits Δ of the numerical solution at the endpoint T (i.e., we write the maximum norm of the error at $t=T$ in the form $10^{-\Delta}$). The computational costs are proportional to mL/h, where h is the fixed steplength, $L := T - t_0$ is the length of the integration interval, and m is the fixed number of outer iterations per step. In actual applications of these methods, some strategy is needed to select h and m. However, since our test problems are such that the exact solution is equally smooth in the whole integration interval, it is reasonable to use fixed h and m.

5.1. Test problems. We briefly discuss a few test problems partly taken from the literature and partly constructed in order to test some special aspect of the methods. All problems are defined on the interval $[t_0, T]$.

Our first problem is the stability test problem of Prothero and Robinson [21]

$$\text{(5.1a)} \qquad \frac{dy}{dt} = -\varepsilon^{-1}(y - g(t)) + g'(t), \qquad y(t_0) = g(t_0), \qquad t_0 = 0, \qquad T = 1,$$

where the exact solution equals $g(t)$ and ε is a small parameter. Prothero and Robinson used this problem to show the order reduction of RK methods when ε is small. In our experiments we set

$$\text{(5.1b)} \qquad g(t) = \cos(t), \qquad \varepsilon = 10^{-3}.$$

The second test problem is the nonlinearization of problem (5.1):

$$\text{(5.2a)} \qquad \frac{dy}{dt} = -\varepsilon^{-1}(y^3 - g(t)^3) + g'(t), \qquad y(t_0) = g(t_0), \qquad t_0 = 0, \qquad T = 1,$$

with exact solution $y(t) = g(t)$ for all values of the parameter ε. As in the preceding problem we set

$$\text{(5.2b)} \qquad g(t) = \cos(t), \qquad \varepsilon = 10^{-3}.$$

The third test problem is that of Kaps [17]:

$$\text{(5.3)} \qquad \begin{gathered} \frac{dy_1}{dt} = -(2+\varepsilon^{-1})y_1 + \varepsilon^{-1}(y_2)^2, \qquad \frac{dy_2}{dt} = y_1 - y_2(1+y_2), \\ y_1(t_0) = y_2(t_0) = 1, \qquad t_0 = 0, \qquad T = 1, \end{gathered}$$

with the smooth exact solution $y_1 = \exp(-2t)$ and $y_2 = \exp(-t)$ for all values of the parameter ε. This problem belongs to the class of problems for which stiffly accurate RK methods do not suffer order reduction whatever small ε is (cf. Hairer, Lubich, and Roche [11]).

The test set of Enright, Hull, and Lindberg [10] contains the following system of ODEs describing a chemical reaction:

$$\text{(5.4a)} \qquad \frac{d\mathbf{y}}{dt} = -\begin{pmatrix} .013 + 1000y_3 & 0 & 0 \\ 0 & 2500y_3 & 0 \\ .013 & 0 & 1000y_1 + 2500y_2 \end{pmatrix}\mathbf{y},$$

with $\mathbf{y}(0) = (1, 1, 0)^T$. Since we use fixed stepsizes in our experiments, we avoided the initial phase by choosing the starting point at $t_0 = 1$ and we used the corresponding

initial values

$$\mathbf{y}(1) \approx \begin{pmatrix} 0.990731920827 \\ 1.009264413846 \\ -.366532612659 \times 10^{-5} \end{pmatrix}. \tag{5.4b}$$

At $t = T = 51$ we found the approximate solution

$$\mathbf{y}(51) = \begin{pmatrix} 0.591045966680 \\ 1.408952165382 \\ -.186793736719 \times 10^{-5} \end{pmatrix}.$$

In order to show the performance of the methods on PDEs we included the convection-diffusion problem

$$\frac{\partial u}{\partial t} = u \frac{\partial^2 u}{\partial x^2} - x \cos(t) \frac{\partial u}{\partial x} - x^2 \sin(t), \qquad 0 \leqq x \leqq 1, \qquad t_0 = 0, \qquad T = 1, \tag{5.5}$$

with Dirichlet boundary conditions and with exact solution $u(x, t) = x^2 \cos(t)$. Standard finite difference discretization of the spatial derivatives on a uniform grid with meshsize $1/40$ leads to a system of 39 ODEs whose exact solution is given by $(j/40)^2 \cos(t)$, $j = 1, \cdots, 39$.

5.2. Effective orders of the correctors. First of all, we want to show that in many stiff problems the property of superconvergence does not pay because of the phenomenon of order reduction, and that strong stability properties may improve the accuracy considerably.

The Tables 5.1(a) and 5.1(b) present Δ values for the various test problems obtained for $L/h = 1, 2, 4, 8, 16$ by iterating the corrector to convergence. From these results we can derive for each test problem the effective orders by computing $(\Delta(h) - \Delta(2h))/0.3$. For h we chose the smallest value for which results are available. The resulting effective

TABLE 5.1(a)

Problems (5.1), (5.2), *and* (5.3) *with* $\varepsilon = 10^{-3}$. *Values of* Δ *for* $L/h = 1, 2, 4, 8, 16$.

Corrector	p	r	k	(5.1)					(5.2)					(5.3) with $\varepsilon = 10^{-3}$				
(4.3a)	4	3	2	4.7	5.4	6.0	6.7	7.7	4.7	5.3	5.9	6.6	7.5	3.3	4.3	5.1	5.9	7.0
(4.4a)	3	3	2	5.1	5.9	6.8	7.8	8.8	5.0	5.8	6.7	7.7	8.7	2.7	3.6	4.4	5.3	6.2
(4.5a)	4	2	2	1.9	2.5	3.1	3.8	4.7	1.9	2.5	3.1	3.8	4.6	1.2	1.8	2.4	3.2	4.3
(4.6a)	3	2	2	4.2	4.7	5.3	5.9	6.5	4.2	4.7	5.2	5.8	6.4	2.4	3.2	4.1	5.0	5.9
(4.7a)	4	4	3	6.1	7.3	8.5	9.7	-	6.0	7.3	8.5	9.7	-	4.2	5.4	6.6	7.8	-
(4.8a)	6	4	3	6.1	7.3	8.6	9.8	-	6.1	7.3	8.5	9.7	-	4.7	6.0	7.3	9.3	-
(4.9a)	4	4	3	6.5	7.6	8.8	10.1	-	6.5	7.6	8.8	10.0	-	3.8	5.0	6.1	7.3	-
(4.10a)	5	3	3	5.0	6.0	6.9	7.9	-	4.9	5.9	6.9	7.8	-	4.0	5.3	6.3	7.3	-
(4.11a)	6	5	4	7.0	8.2	9.5	-	-	6.9	8.1	9.4	-	-	5.4	6.7	8.0	-	-
(4.12a)	8	5	4	7.1	8.4	9.6	-	-	7.0	8.3	9.5	-	-	5.6	6.8	8.2	-	-
(4.13a)	5	5	4	7.5	8.9	10.5	-	-	7.4	8.9	10.4	-	-	5.8	7.2	8.8	-	-
(4.14a)	7	4	4	6.3	7.4	8.6	-	-	6.3	7.3	8.5	-	-	5.0	6.4	7.8	-	-

TABLE 5.1(b)

Problems (5.3) *with* $\varepsilon = 10^{-8}$, (5.4) *and* (5.5). *Values of* Δ *for* $L/h = 1, 2, 4, 8, 16$.

Corrector	p	r	k	(5.3) with $\varepsilon = 10^{-8}$					(5.4)					(5.5)				
(4.3a)	4	3	2	3.3	4.5	5.7	6.9	8.1	4.5	5.7	6.9	8.2	9.4	3.2	4.2	5.4	6.5	7.7
(4.4a)	3	3	2	2.7	3.6	4.4	5.3	6.2	3.1	4.0	4.9	5.8	6.7	3.1	4.0	4.8	5.7	6.6
(4.5a)	4	2	2	1.2	1.8	2.4	3.0	3.6	5.0	6.1	7.3	8.5	9.7	1.9	2.6	3.2	3.9	4.8
(4.6a)	3	2	2	2.4	3.2	4.1	5.0	5.9	3.4	4.3	5.2	6.1	7.0	2.5	3.2	4.0	4.8	5.7
(4.7a)	4	4	3	4.2	5.4	6.7	7.9	-	4.7	5.9	7.1	8.3	-	4.6	5.9	7.2	8.4	-
(4.8a)	6	4	3	5.4	7.2	9.0	10.8	-	6.4	8.3	10.1	11.8	-	4.8	6.2	7.7	9.1	-
(4.9a)	4	4	3	3.9	5.0	6.2	7.3	-	4.2	5.4	6.6	7.8	-	4.5	5.6	6.8	7.9	-
(4.10a)	5	3	3	4.4	5.8	7.3	8.8	-	5.3	6.8	8.3	9.8	-	3.6	4.8	6.1	7.3	-
(4.11a)	6	5	4	5.9	7.7	9.6	-	-	6.7	8.5	10.3	-	-	5.7	7.4	9.2	-	-
(4.12a)	8	5	4	7.8	10.2	12.6	-	-	8.6	11.0	-	-	-	6.0	7.7	9.5	-	-
(4.13a)	5	5	4	6.0	7.4	8.8	-	-	6.9	8.2	9.7	-	-	6.4	7.8	9.3	-	-
(4.14a)	7	4	4	6.6	8.7	10.8	-	-	7.9	9.8	11.8	-	-	5.2	6.5	8.0	-	-

orders are listed in Table 5.2. For each problem, the result of the most accurate corrector is indicated in bold face.

The results for the first three problems clearly demonstrate that the various methods often do not show their step point order, so that the property of superconvergence is of limited value in the case of stiff problems.

5.3. Performance of the iteration process for two-processor correctors. In this section, we consider the performance of the iteration method for solving the two-processor corrector equations. Since the rate of convergence of a particular iteration method turned out to be comparable for the Newton-Cotes corrector and the Lagrange corrector, we only present results for the most accurate one. In the case of the Gauss and Radau corrector, the iteration methods behaved quite differently so that we include

TABLE 5.2

Effective orders shown by the correctors for problems (5.1)-(5.5).

Corrector	p	r	k	(5.1)	(5.2)	(5.3) $\varepsilon = 10^{-3}$	(5.3) $\varepsilon = 10^{-8}$	(5.4)	(5.5)
Newton-C. (4.3a)	4	3	2	3.3	3.0	**3.7**	**4.0**	4.0	**4.0**
Lagrange (4.4a)	3	3	2	**3.3**	**3.3**	3.0	3.0	3.0	3.0
Gauss (4.5a)	4	2	2	3.0	2.7	3.7	2.0	**4.0**	3.0
Radau IIA (4.6a)	3	2	2	2.0	2.0	3.0	3.0	3.0	3.0
Newton-C. (4.7a)	4	4	3	4.0	4.0	4.0	4.0	4.0	4.0
Lobatto IIIA (4.8a)	6	4	3	4.0	4.0	**6.7**	**6.0**	**5.7**	**4.7**
Lagrange (4.9a)	4	4	3	**4.3**	**4.0**	4.0	3.7	4.0	3.7
Radau IIA (4.10a)	5	3	3	3.3	3.0	3.3	5.0	5.0	4.0
Newton-C. (4.11a)	6	5	4	4.3	4.3	4.3	6.3	6.0	6.0
Lobatto IIIA (4.12a)	8	5	4	4.0	4.0	4.7	**8.0**	**8.0**	**6.0**
Lagrange (4.13a)	5	5	4	**5.3**	**5.0**	**5.3**	4.7	5.0	5.0
Radau IIA (4.14a)	7	4	4	4.0	4.0	4.7	7.0	6.7	5.0

results for both correctors. Moreover, the Gauss corrector was also iterated with a matrix D defined by the relation (3.12). Since for the two-processor Gauss corrector (4.5a) we have $A\mathbf{e}=\mathbf{c}$, i.e., $A^{-1}\mathbf{c}=\mathbf{e}$, it follows that $D^{-1}\mathbf{c}=\mathbf{e}$, so that

$$D=\operatorname{diag}(\mathbf{c})=\frac{1}{12}\begin{pmatrix}6-2\sqrt{3} & 0\\ 0 & 6+2\sqrt{3}\end{pmatrix},\qquad \rho(I-D^{-1}A)=0.5.$$

In the Tables 5.3 and 5.4 we only present results for the problems (5.2) and (5.4) for which most methods, respectively, show their stage order and their step point order (additional results for the other test problems may be found in the Appendix to [14]). Divergence of the inner iteration is indicated by *, and values in bold indicate that the accuracy of the corrector is reached (and that Δ does not change anymore). For several values of L/h the accuracies corresponding to the correctors of Lagrange type (first column), of Gauss-Legendre with D defined by (4.5b) (second column), of Gauss-Legendre with D defined above (third column), and of Radau IIA (fourth column) are listed. These results confirm that, in general, the Gauss corrector is not suited to be iterated by diagonal iteration methods when started with an explicit predictor.

5.4. Efficiency of diagonally iterated IRK correctors. In this final section, we compare the efficiency of the diagonally iterated IRK correctors with three fourth-order

TABLE 5.3

Values of Δ for problem (5.2) *obtained by iterating the Lagrange corrector* (4.4a), *Gauss corrector* (4.5a), *and Radau* IIA *corrector* (4.6a).

m	$L/h=1$				$L/h=2$				$L/h=4$				$L/h=8$				$L/h=16$			
1	3.5	0.3	0.3	3.8	4.1	−2.2	−2.2	5.3	4.0	*	*	4.8	3.6	*	*	5.0	2.7	*	*	5.3
2	**5.0**	1.0	0.7	**4.2**	**5.8**	−1.1	1.1	**4.7**	6.5	*	1.1	**5.2**	6.7	*	0.6	5.9	6.7	*	*	6.7
3		**1.9**	1.0			2.4	2.2		**6.7**	2.9	2.6		**7.7**	3.9	3.2	**5.8**	8.4	1.9	3.8	**6.4**
4			1.4			**2.5**	1.9			**3.1**	2.5			**3.8**	3.1		**8.7**	**4.6**	3.8	
5			1.8				2.1				2.6				3.2				3.9	
⋮			⋮				⋮				⋮				⋮				⋮	
10			**1.9**				2.6				3.2				4.1				5.3	
20							**2.5**				**3.1**				**3.8**				**4.6**	

TABLE 5.4

Values of Δ for Problem (5.4) *obtained by iterating the Lagrange corrector* (4.4a), *Gauss corrector* (4.5a), *and Radau* IIA *corrector* (4.6a).

m	$L/h=1$				$L/h=2$				$L/h=4$				$L/h=8$				$L/h=16$			
1	2.1	1.2	1.2	1.7	2.3	1.5	1.5	2.1	2.6	*	*	2.4	2.8	*	*	2.7	3.1	*	*	3.0
2	3.4	2.6	2.2	2.9	3.9	2.9	2.8	3.5	4.5	*	3.4	4.1	5.2	*	4.0	4.7	5.8	*	4.5	5.3
3	4.3	3.8	3.0	3.6	5.4	4.8	3.8	4.5	6.4	5.7	4.7	5.4	7.4	6.6	5.6	6.3	8.3	6.9	6.5	7.2
4	**4.5**	4.7	3.8	**3.4**	**5.7**	5.9	4.9	**4.3**	**6.9**	7.1	6.1	**5.2**	8.1	8.3	7.2	**6.1**	9.3	9.5	8.4	**7.0**
5		**5.0**	4.4			**6.1**	5.7			**7.3**	7.1		**8.2**	**8.5**	8.3		**9.4**	**9.7**	9.5	
6			4.8				**6.1**				**7.3**				**8.5**				9.5	
7			**5.0**																**9.7**	

DIRK methods from the literature, viz. the three-stage method generated by the Butcher array:

$$
\begin{array}{c|ccc}
\frac{1}{2}(1+\xi) & \frac{1}{2}(1+\xi) & & \\
\frac{1}{2} & -\frac{1}{2}\xi & \frac{1}{2}(1+\xi) & \\
\frac{1}{2}(1-\xi) & (1+\xi) & -(1+2\xi) & \frac{1}{2}(1+\xi) \\
\hline
 & \frac{1}{6\xi^2} & 1-\frac{1}{3\xi^2} & \frac{1}{6\xi^2}
\end{array}, \qquad \xi=\frac{2}{3}\sqrt{3}\cos\left(\frac{\pi}{18}\right) \tag{5.6}
$$

(cf. Crouzeix [8] and Alexander [1]), and the four-stage, parallel DIRK methods of Iserles and Nørsett [16]:

$$
\begin{array}{c|cccc}
\frac{1}{2} & \frac{1}{2} & & & \\
1 & 0 & 1 & & \\
\frac{1}{2} & \frac{3}{2} & -\frac{3}{2} & \frac{1}{2} & \\
0 & -3 & 2 & 0 & 1 \\
\hline
 & \frac{1}{3} & \frac{1}{6} & \frac{1}{3} & \frac{1}{6}
\end{array}, \tag{5.7}
$$

$$
\begin{array}{c|cccc}
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{2}{3} & 0 & \frac{2}{3} & & \\
\frac{1}{2} & -\frac{5}{2} & \frac{5}{2} & \frac{1}{2} & \\
\frac{1}{3} & -\frac{5}{3} & \frac{4}{3} & 0 & \frac{2}{3} \\
\hline
 & -1 & \frac{3}{2} & -1 & \frac{3}{2}
\end{array}. \tag{5.8}
$$

The method (5.6) is A-stable and requires three sequential stages per step. The methods (5.7) and (5.8) are A-stable and L-stable, respectively, and require only two sequential stages per step (when run on a two-processor computer).

We restrict our considerations to the above three DIRK methods and to the Newton–Cotes, Lobatto IIIA, Lagrange, and Radau IIA correctors where each method uses a fixed number of m iterations per step. Recalling that iterating an IRK corrector by means of m diagonal iterations in each step yields a method that is in fact a DIRK method with m sequential stages, we conclude that all methods have in common that they belong to the class of DIRK methods. However, in the case of the "genuine" DIRK methods (5.6), (5.7), and (5.8), the number of sequential stages per step is

known in advance, whereas in the case of the DIRK methods based on iteration the number of sequential stages m that yields acceptable accuracies, is not known in advance and, in actual computation, it should be determined on the basis of some local error strategy. On the other hand, as we shall see, the accuracy of the iterated methods is less sensitive to the phenomenon of order reduction.

In the Tables 5.5 and 5.6, m always denotes the number of sequential stages per step. Hence, all results in one column of these tables correspond to DIRK methods that use m sequential stages per step, so that all results corresponding to the same value of mL/h required roughly the same computational effort. In the tables, the highest value of Δ corresponding to the same mL/h value, that is, the "most efficient"

TABLE 5.5(a)

Problem (5.2): *results for diagonally iterated correctors and for the methods* (5.6), (5.7), *and* (5.8).

Method	k	L/h	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$	$\cdots m=\infty$
Crouzeix-Alex. (5.6)	1	1	-	-	1.0								-
Iserles-Nørsett (5.7)	2		-	1.5									-
Iserles-Nørsett (5.8)			-	2.1									-
Newton-C. (4.3)			3.4	4.7									4.7
Lagrange (4.4)			3.5	**5.0**									5.0
Radau IIA (4.6)			**3.8**	4.2									4.2
Newton-C. (4.7)	3		3.2	3.7	**5.6**	6.1	6.0						6.0
Lobatto IIIA (4.8)			3.0	2.7	4.7	6.0	6.0	6.1					6.1
Lagrange (4.9)			3.2	3.9	5.5	**6.7**	**6.5**						6.5
Radau IIA (4.10)			3.4	3.1	5.0	4.9							4.9
Newton-C. (4.11)	4		3.1	3.6	4.9	4.7	5.2	6.0	**7.2**	7.0	6.9		6.9
Lobatto IIIA (4.12)			2.7	2.2	2.3	3.9	4.6	5.4	6.8	6.9	7.0		7.0
Lagrange (4.13)			3.0	2.8	3.1	3.9	5.0	6.4	7.1	**7.3**	**7.4**		**7.4**
Radau IIA (4.14)			2.9	2.8	3.0	4.7	5.6	**6.8**	6.3				6.3
Crouzeix-Alex. (5.6)	1	2	-	-	2.5								-
Iserles-Nørsett (5.7)	2		-	2.4									-
Iserles-Nørsett (5.8)			-	2.7									-
Newton-C. (4.3)			4.0	5.3									5.3
Lagrange (4.4)			4.1	**5.8**									5.8
Radau IIA (4.6)			**5.3**	4.7									4.7
Newton-C. (4.7)	3		3.4	3.5	**6.4**	**8.1**	7.2	7.3					7.3
Lobatto IIIA (4.8)			3.0	2.2	5.3	6.0	7.3						7.3
Lagrange (4.9)			3.5	3.8	5.9	7.5	**7.6**						7.6
Radau IIA (4.10)			3.8	2.8	5.9	5.7	5.9						5.9
Newton-C. (4.11)	4		3.3	3.3	5.2	5.2	5.3	5.9	6.7	7.8	8.3	8.1	8.1
Lobatto IIIA (4.12)			2.3	1.1	1.4	4.0	4.5	5.5	6.9	7.3	8.4	8.3	8.3
Lagrange (4.13)			2.9	2.3	2.7	4.9	5.2	6.5	**8.3**	**8.9**			**8.9**
Radau IIA (4.14)			2.8	2.2	2.6	5.0	6.0	7.0	7.5	7.3			7.3
Crouzeix-Alex. (5.6)	1	4	-	-	2.8								-
Iserles-Nørsett (5.7)	2		-	3.0									-
Iserles-Nørsett (5.8)			-	3.2									-
Newton-C. (4.3)			3.9	5.8	5.9								5.9
Lagrange (4.4)			4.0	**6.5**	**6.7**								6.7
Radau IIA (4.6)			**4.8**	5.2									5.2
Newton-C. (4.7)	3		3.1	3.0	6.6	**7.7**	8.4	8.5					8.5
Lobatto IIIA (4.8)			2.3	0.7	5.5	6.2	7.7	8.1	8.5				8.5
Lagrange (4.9)			3.2	3.5	6.2	**7.7**	**9.9**	**8.8**					8.8
Radau IIA (4.10)			3.6	2.0	5.6	6.2	6.8	6.9					6.9
Newton-C. (4.11)	4		2.9	2.5	5.0	5.5	5.5	6.0	6.8	7.7	8.7	9.8	9.4
Lobatto IIIA (4.12)			1.1	*	*	5.0	4.3	5.6	6.4	7.2	8.3	9.0	9.5
Lagrange (4.13)			2.3	0.8	1.5	5.1	5.6	6.8	7.9	**8.8**	**9.7**	**10.8**	**10.4**
Radau IIA (4.14)			2.1	0.6	1.2	5.2	6.3	7.9	8.4	8.5			8.5

TABLE 5.5(b)

Problem (5.2): *efficiency test of fourth-order methods.*

Method	p	m	k	$mL/h=4$	$mL/h=8$	$mL/h=16$
Iserles–Nørsett (5.7)	4	2	2	2.4	3.0	3.6
Iserles–Nørsett (5.8)	4	2	2	2.7	3.2	3.8
Newton-C. (4.3)	4	4	2	4.7	5.3	5.9
Newton-C. (4.7)	4	4	3	6.1	**8.1**	**7.7**
Lobatto IIIA (4.8)	4	4	3	6.0	6.0	6.2
Lagrange (4.9)	4	4	3	**6.7**	7.5	**7.7**
Radau IIA (4.10)	4	4	3	4.9	5.7	6.2
Newton-C. (4.11)	4	4	4	4.7	5.2	5.5
Lobatto IIIA (4.12)	4	4	4	3.9	4.0	5.0
Lagrange (4.13)	4	4	4	3.9	4.9	5.1
Radau IIA (4.14)	4	4	4	4.7	5.0	5.2

integration result, is indicated in bold. As in the preceding section, we only present results for the problems (5.2) and (5.4). Results for the additional test problems may be found in the Appendix to [14].

In the case of the nonlinear Prothero–Robinson problem, Table 5.5(a) shows that the number of iterations needed by the iterated methods to "reach" the accuracy of the corrector solution increases with k, that is, the higher-order methods need more iterations to solve the corrector; moreover, they have a "slow start": after two iterations the accuracy is still rather modest, whereas the lower-order methods have already converged, showing full corrector-precision. This can be explained by observing that we used a zero-order predictor for $\mathbf{Y}^{(0)}$ for all k, so that the "distance" between predictor and corrector solution increases with k. Thus, for this problem, the lower-order methods are more efficient than the higher-order ones, unless very high accuracies are requested. Furthermore, when we compare the various types of iterated methods (Newton–Cotes, Lobatto, Lagrange, or Radau), then the Lobatto IIIA methods perform not as well whereas the strongly A-stable Lagrange methods are slightly superior to the others. In the case of the "genuine" DIRK methods (5.6), (5.7) and (5.8), the Iserles–Nørsett methods are more accurate than the Crouzeix–Alexander method, which is presumably due to the L-stability property of the Iserles–Nørsett method.

It is of particular interest to see how the iterated methods compare with the "genuine" DIRK methods. For example, Table 5.5(a) shows that the Newton–Cotes, Lobatto IIIA, Lagrange and Radau IIA based methods, respectively, produce 5, 0, 21 and 4 "most efficient" results, whereas the "genuine" DIRK methods none. A further indication of the superiority of the iterated methods is given by Table 5.5(b) where we list results for the iterated methods with $m=4$ and for the parallel DIRK methods (5.7) and (5.8). All these methods have step point order $p=4$, but the accuracies obtained for the same computational-costs value of mL/h differ largely, which is caused by the order reduction exhibited by the "genuine" DIRK methods.

For the more innocent chemical reaction problem (5.4) the order reduction is not shown. Table 5.6(a) shows that the high-order iterated methods again require more iterations to obtain the corrector precision than the lower-order methods, however, here for low values of m, all iterated methods are roughly equally efficient. Furthermore, the scores of "most efficient" results for the Newton–Cotes, Lobatto IIIA, Lagrange, and Radau IIA based methods are, respectively, 8, 5, 6, and 7, and among the DIRK methods only (5.7) scores twice. The analogue of Table 5.5(b) is given by Table 5.6(b).

TABLE 5.6(a)

Problem (5.4): *results for diagonally iterated correctors and for the methods* (5.6), (5.7), *and* (5.8).

Method	k	L/h	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$	$\cdots$ $m=\infty$
Crouzeix-Alex. (5.6)	1	1	-	-	3.4								-
Iserles-Nørsett (5.7)	2		-	3.4									-
Iserles-Nørsett (5.8)			-	3.3									-
Newton-C. (4.3)			**2.1**	3.4	4.3	4.5							4.5
Lagrange (4.4)			**2.1**	3.5	3.1								3.1
Radau IIA (4.6)			1.7	2.9	3.6	3.4							3.4
Newton-C. (4.7)	3		1.8	3.5	5.1	4.7							4.7
Lobatto IIIA (4.8)			1.6	3.1	4.3	5.6	6.3	6.4					6.4
Lagrange (4.9)			1.8	3.5	4.3	4.2							4.2
Radau IIA (4.10)			2.0	3.2	4.3	5.9	5.3						5.3
Newton-C. (4.11)	4		1.7	**3.6**	5.2	6.5	6.7						6.7
Lobatto IIIA (4.12)			1.4	2.7	4.6	6.0	7.1	**8.3**	**8.6**				**8.6**
Lagrange (4.13)			1.6	3.1	**5.8**	6.6	7.0	6.9					6.9
Radau IIA (4.14)			1.5	3.2	4.8	**7.4**	**7.8**	7.9					7.9
Crouzeix-Alex. (5.6)	1	2	-	-	4.4								-
Iserles-Nørsett (5.7)	2		-	**4.5**									-
Iserles-Nørsett (5.8)			-	4.4									-
Newton-C. (4.3)			**2.3**	3.9	5.4	5.7							5.7
Lagrange (4.4)			**2.3**	**4.5**	4.0								4.0
Radau IIA (4.6)			2.1	3.5	4.5	4.3							4.3
Newton-C. (4.7)	3		2.0	4.2	6.2	5.9							5.9
Lobatto IIIA (4.8)			1.9	3.8	5.1	6.8	8.1	8.3					8.3
Lagrange (4.9)			2.1	4.1	5.5	5.4							5.4
Radau IIA (4.10)			2.2	3.8	5.1	6.9	6.8						6.8
Newton-C. (4.11)	4		2.0	**4.5**	**6.7**	7.9	8.5						8.5
Lobatto IIIA (4.12)			1.7	3.3	5.4	7.2	8.5	10.0	**10.9**	**11.0**			**11.0**
Lagrange (4.13)			1.9	3.7	6.3	7.5	8.3	8.2					8.2
Radau IIA (4.14)			1.8	3.7	5.6	**8.0**	**8.8**	**10.1**	9.8				9.8
Crouzeix-Alex. (5.6)	1	4	-	-	5.5								-
Iserles-Nørsett (5.7)	2		-	**5.7**									-
Iserles-Nørsett (5.8)			-	5.6									-
Newton-C. (4.3)			**2.6**	4.5	6.4	6.9							6.9
Lagrange (4.4)			**2.6**	4.7	4.9								4.9
Radau IIA (4.6)			2.4	4.1	5.4	5.2							5.2
Newton-C. (4.7)	3		2.3	5.0	**7.2**	7.1							7.1
Lobatto IIIA (4.8)			2.2	4.4	6.0	7.9	9.7	10.1					10.1
Lagrange (4.9)			2.4	4.8	6.8	6.6							6.6
Radau IIA (4.10)			2.5	4.5	6.0	7.9	8.3						8.3
Newton-C. (4.11)	4		2.3	5.4	7.1	8.9	**10.6**	10.3					10.3
Lobatto IIIA (4.12)			2.0	4.0	6.1	8.4	10.1	11.9	**12.3**				**12.3**
Lagrange (4.13)			2.2	4.2	**7.2**	8.7	9.9	9.7					9.7
Radau IIA (4.14)			2.1	4.3	6.6	**9.1**	10.2	**12.2**	11.8				11.8

It reveals that the iterated methods are usually much more efficient than the parallel DIRK methods, and in any case they are at least competitive.

6. Concluding remarks. In this paper we have derived a diagonally implicit iteration scheme to solve a fully implicit Runge-Kutta method. The structure of this iteration process is such that a parallel computer can be fully exploited. Starting with an implicit RK method with k implicit stages (the corrector), each iteration requires the solution of k systems of equations of dimension equal to the number of ODEs. Since these systems can be solved completely independently, the *effective* computational work per iteration equals the solution of one such system, provided that k processors are available.

TABLE 5.6(b)

Problem (5.4): *efficiency test of fourth-order methods.*

Method	p	m	k	$mL/h=4$	$mL/h=8$	$mL/h=16$
Iserles–Nørsett (5.7)	4	2	2	4.5	5.7	6.9
Iserles–Nørsett (5.8)	4	2	2	4.4	5.6	6.7
Newton-C. (4.3)	4	4	2	4.5	5.7	6.9
Newton-C. (4.7)	4	4	3	4.7	5.9	7.1
Lobatto IIIA (4.8)	4	4	3	5.6	6.8	7.9
Lagrange (4.9)	4	4	3	4.2	5.4	6.6
Radau IIA (4.10)	4	4	3	5.9	6.9	7.9
Newton-C. (4.11)	4	4	4	6.5	7.9	8.9
Lobatto IIIA (4.12)	4	4	4	6.0	7.2	8.4
Lagrange (4.13)	4	4	4	6.6	7.5	8.7
Radau IIA (4.14)	4	4	4	**7.4**	**8.0**	**9.1**

The free parameters in the iteration scheme are chosen in such a way that the corresponding stability functions converge as quickly as possible to the stability function of the corrector, which is chosen to be (at least) A-acceptable. Although we have numerical evidence that this is not a bad choice, we do not claim that it is the best possible. In a forthcoming paper it is intended to give theoretical support for this choice.

A second aspect considered in this paper, is the choice of the particular corrector method. The well-known implicit RK methods of high classical order, such as the Gauss–Legendre, Radau, and Lobatto methods, seem to be suitable candidates. However, since it is the *stage order* which usually determines the order behaviour in integrating stiff differential equations, these methods are not necessarily optimal correctors. Because the stage order is significantly smaller than the classical order for these methods, we will encounter the phenomenon of order reduction. Therefore, we also considered Newton–Cotes and Lagrange correctors, which have—for the same number of implicit relations per iteration—a stage order which is one higher than for Gauss–Legendre and Radau methods and is equal to the stage order of Lobatto methods.

Apart from these order considerations, it turned out that the stability behaviour of the iterated scheme largely depends on the choice of the corrector. For example, it is shown that the Gauss–Legendre corrector is not suitable in this context, since it is not stiffly accurate. Consequently, only for very "innocent" stiff problems, where we have no order reduction, the Gauss–Legendre corrector is useful, but as a method for general stiff problems it is disadvantageous.

The other four types of correctors are all stiffly accurate, which has the effect that certain classes of stiff problems can be integrated without order reduction. For such problems the classical order should be a decisive factor, viz. in these cases the Lobatto IIIA corrector is superior and also the Newton–Cotes corrector is a good choice. However, these correctors are only A-stable and it is shown that the stability function of the iterated method is not A-acceptable unless the corrector is really solved. This means that the iteration process based on these correctors easily encounters stability problems. Hence, a corrector possessing better stability characteristics, such as the Radau IIA method (L-stable) and the Lagrange method (strongly A-stable), will be much more robust. We showed that after a few iterations the stability function of the iterated methods based on these correctors is A-acceptable.

Since the stage order of the Lagrange corrector is one larger than that of the Radau IIA corrector, we think that it is a good choice for integrating general stiff

equations; it combines adequate stability characteristics with a relatively high stage order. Our numerical experiments confirm this advice.

Furthermore, we have compared our methods with sequential and parallel DIRK methods from the literature. This comparison is rather obvious since the effective computational work per iteration equals the work per stage in a DIRK method. It turned out that the diagonally iterated RK methods are much more efficient than the "conventional" DIRKs. The reason is that only low order "conventional" DIRKs with good stability properties are available in the literature and, more importantly, these DIRKs have a stage order equal to 1. This property gives these methods a very poor performance in case of general stiff problems.

Finally, we remark that the construction of diagonally iterated methods of arbitrarily high order is straightforward, and we observed in our experiments that, especially the high order methods, showed remarkably high accuracies.

Acknowledgment. The authors are grateful to Dr. W. H. Hundsdorfer for carefully reading the manuscript and for many valuable suggestions during the investigations reported in this paper.

REFERENCES

[1] R. ALEXANDER, *Diagonally implicit Runge-Kutta methods for stiff* ODEs, SIAM J. Numer. Anal., 14 (1977), pp. 1006-1021.

[2] K. BURRAGE, *A special family of Runge-Kutta methods for solving stiff differential equations*, BIT, 18 (1978), pp. 22-41.

[3] ———, *The error behaviour of a general class of predictor-corrector methods*, CMSR Report, University of Liverpool, Liverpool, U.K., 1989.

[4] K. BURRAGE, P. J. VAN DER HOUWEN, W. H. HUNDSDORFER, AND B. P. SOMMEIJER, *Diagonal iteration of Runge-Kutta methods*, in preparation.

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

[6] ———, *A transformed implicit Runge-Kutta method*, J. Assoc. Comput. Mach., 26 (1979), pp. 731-738.

[7] ———, *The Numerical Analysis of Ordinary Differential Equations, Runge-Kutta and General Linear Methods*, John Wiley, New York, 1987.

[8] M. CROUZEIX, *Sur l'approximation des équations différentielles opérationnelles linéaires par des méthodes de Runge-Kutta*, Ph.D. thesis, Université de Paris, Paris, France, 1975.

[9] K. DEKKER AND J. G. VERWER, *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations*, CWI Monograph 2, North-Holland, Amsterdam, 1984.

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

[11] E. HAIRER, CH. LUBICH, AND M. ROCHE, *Error of Runge-Kutta methods for stiff problems studied via differential algebraic equations*, BIT, 28 (1988), pp. 678-700.

[12] P. J. VAN DER HOUWEN AND B. P. SOMMEIJER, *Parallel iteration of high-order Runge-Kutta methods with stepsize control*, J. Comput. Appl. Math. 29 (1990), pp. 111-127.

[13] P. J. VAN DER HOUWEN, B. P. SOMMEIJER, AND W. COUZY, *Embedded diagonally implicit Runge-Kutta algorithms on parallel computers*, Report NM-R8912, Centre for Mathematics and Computer Science, Amsterdam, the Netherlands, Math. Comp., 1992, to appear.

[14] P. J. VAN DER HOUWEN AND B. P. SOMMEIJER, *Iterated Runge-Kutta methods on parallel computers*, Report NM-R9001, Centre for Mathematics and Computer Science, Amsterdam, the Netherlands, 1990.

[15] W. H. HUNDSDORFER, private communication, 1990.

[16] A. ISERLES AND S. P. NØRSETT, *On the theory of parallel Runge-Kutta methods*, IMA J. Numer. Anal., 10 (1990), pp. 463-488.

[17] P. KAPS, *Rosenbrock-type methods*, in Numerical Methods for Stiff Initial Value Problems, G. Dahlquist and R. Jeltsch, eds., Bericht nr. 9, Inst. für Geometrie und Praktische Mathematik der RWTH Aachen, Aachen, Germany, 1981.

[18] I. LIE, *Some aspects of parallel Runge-Kutta methods*, Report No. 3/87, Division Numerical Mathematics, University of Trondheim, Trondheim, Norway, 1987.

[19] S. P. NØRSETT, *Semi-explicit Runge-Kutta methods*, Report Mathematics and Computation No. 6/74 Department of Mathematics, University of Trondheim, Trondheim, Norway, 1974.

[20] S. P. NØRSETT AND H. H. SIMONSEN, *Aspects of parallel Runge-Kutta methods*, in Numerical Method for Ordinary Differential Equations, A. Bellen, C. W. Gear, and E. Russo, eds., Proc. L'Aquil 1987, Lecture Notes in Mathematics, Vol. 1386, Springer-Verlag, Berlin, 1989.

[21] A. PROTHERO AND A. ROBINSON, *On the stability and accuracy of one-step methods for solving sti systems of ordinary differential equations*, Math. Comp., 28 (1974), pp. 145–162.

[22] B. P. SOMMEIJER, W. COUZY, AND P. J. VAN DER HOUWEN, *A-stable parallel block methods*, Repo NM-R8918, Centre for Mathematics and Computer Science, Amsterdam, the Netherlands, App Numer. Math., to appear.

[23] H. A. WATTS AND L. F. SHAMPINE, *A-stable block implicit one-step methods*, BIT, 12 (1972), pp. 252–26