Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

P.J. van der Houwen, B.P. Sommeijer

Iterated Runge-Kutta methods on parallel computers

Department of Numerical Mathematics Report NM-R9001 January

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

Iterated Runge-Kutta Methods on Parallel Computers

P.J. van der Houwen & B.P. Sommeijer

Centre for Mathematics and Computer Science
Post box 4079, 1009 AB Amsterdam, The Netherlands

In this paper, we study 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 we get fast convergence to the stability characteristics of the corrector method. By means of numerical experiments we show that also 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. Moreover, by iterating e.g. Radau correctors, we can construct methods of any order, whereas the order of accuracy of the conventionally constructed DIRK methods is limited to four.

1980 Mathematics Subject Classification: 65M10, 65M20
1982 CR Categories: 5.17
Key Words and Phrases: Diagonally implicit Runge-Kutta methods, parallelism, stability.

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.,

$$\frac{dy(t)}{dt} = f(t, y(t)), \quad y(t_0) = y_0, \quad y : \mathbb{R} \to \mathbb{R}^d, \quad f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d, \tag{1.1}$$

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, **b** and **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, one has proposed diagonally implicit RK methods (DIRK methods) possessing a lower triangular A matrix and therefore requiring (in general) 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 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 Butcher which possess a high stage order and, like DIRK methods, are only diagonally implicit. However, they require, additionally, in each step special transformations so 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 et al. [9]. 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

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

where b_0 is a scalar, **a**, **b** and **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 **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

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

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

$$\mathbf{Y}_{n+1} - hA\mathbf{f}(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_{n+1}) = \mathbf{e}y_n + ha\mathbf{f}(t_n, y_n), \quad y_{n+1} = y_n + hb_0f(t_n, y_n) + h\mathbf{b}^T\mathbf{f}(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_{n+1}). \qquad (1.3')$$

Here, **e** is the vector with unit entries, and we used the convention that for any given vectors $\mathbf{v}=(v_j)$ and $\mathbf{t}=(t_j)$, $\mathbf{f}(\mathbf{t},\mathbf{v})$ denotes the vector with entries $f(t_j,v_j)$. If $b_0=0$ and $\mathbf{a}=\mathbf{0}$, 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 in solving this system, the methods (1.2) and (1.3) may be considered as equally expensive.

The vectors $\mathbf{Y}_{n+1}$ and **c** will respectively be called the *stage vector* and the *block point vector*, and the points t_n and t_n+c_jh 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 **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 [17], and if the components of **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. It is our aim to construct Lagrange methods with better stability properties than Newton-Cotes and Lobatto IIIA methods, i.e., 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

$$b_0 = \mathbf{e}_k^T\mathbf{a}, \quad \mathbf{b}^T = \mathbf{e}_k^TA, \quad c_k = 1, \qquad (1.4)$$

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 et al. [7] 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*), 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 Section 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).

Secondly, it turned out that the iterated methods based on strongly A-stable correctors (such as the Radau IIA correctors and the Lagrange correctors derived in Section 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 by the conventionally constructed DIRK methods for many stiff problems, is not shown by the DIRK methods constructed by diagonal iteration. In a forthcoming paper it is intended to present a theoretical analysis of this phenomenon using the error analysis proposed in Burrage [2].

2. Accuracy and Stability of the Corrector

In the 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 [3], (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.,

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

The stage-order conditions for (1.3') are straightforwardly derived (cf. [16]) and are given by

$$\text{(2.2)} \qquad \mathbf{C}_j = \mathbf{0},\ j = 1, ..., r; \quad \mathbf{C}_1 := \mathbf{a} + A\mathbf{e} - \mathbf{c}; \quad \mathbf{C}_j := jAc^{j-1} - \mathbf{c}^j,\ j = 2, 3, \ldots .$$

Thus, to achieve stage order r for a given block point vector **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'):

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

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

$$\text{(2.4)} \qquad D_j = 0,\ j = 1, ..., q; \quad D_1 := b_0 + \mathbf{b}^T\mathbf{e} - 1; \quad D_j := j\mathbf{b}^T\mathbf{c}^{j-1} - 1,\ j = 2, 3, \ldots$$

are satisfied. We remark that p may be larger than $\min\{r+1,q\}$ if the methods possesses 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

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

Assuming that **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 **c**.

As already observed in the introduction, the usual approach in exploiting the vector **c** is the maximization of the step point order (to obtain 'superconvergence'). Alternatively, one may use **c** for improving the stability of the method or for the minimization of error constants. In this paper, we shall use **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

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

Hence,

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

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

$$(2.8) \qquad y_{n+1} = R(z)y_n, \quad R(z) := e_k^T[I - zA]^{-1}[e + za].$$

We observe that by applying Cramer's rule, the stability functions (2.7) and (2.8) can respectively be rewritten in the form

$$(2.9) \qquad R(z) := 1 + (b_0 - 1)z + z\frac{\det[I - zA + (e + za)b^T]}{\det[I - zA]}, \quad R(z) := -1 + \frac{\det[I - zA + (e + za)e_k^T]}{\det[I - zA]}.$$

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 c are equally spaced, it was shown in Watts and Shampine [17] that they are A-stable for $k \le 8$ (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.

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 [3]
Lobatto IIIA	k+1	2k	k+1	A-stable for all k	yes	Dekker and Verwer [5]
Radau IIA	k	2k-1	k	L-stable for all k	yes	Butcher [3]
Newton-Cotes	k+1	2[(k+2)/2]	k+1	A-stable for $k \le 8$	yes	Watts and Shampine [17]
Lagrange	k+1	k+1	k+1	Strongly A-stable	yes	For $k \le 4$ see Section 4

3. Diagonal Iteration

We shall use a diagonal iteration method to solve the stage vector Y_{n+1} from the fully implicit (corrector) equation defined in (1.3'). For scalar differential equations, the iteration method reads

$$(3.1a) \qquad \begin{aligned} &Y^{(1)} - hDf(et_n + ch, Y^{(1)}) = y_n e + haf(t_n, y_n) + h[A - D]\, f(t^{(0)}, Y^{(0)}), \\ &Y^{(j)} - hDf(et_n + ch, Y^{(j)}) = y_n e + haf(t_n, y_n) + h[A - D]\, f(et_n + ch, Y^{(j-1)}), \quad j = 2, 3, \ldots, \end{aligned}$$

where $(t^{(0)}, Y^{(0)})$ is an initial approximation to $(et_n + ch, Y_{n+1})$ and D is a free diagonal matrix. If m iterations are performed, then y_{n+1} is defined by

$$(3.1b) \qquad y_{n+1} = y_n + hb_0 f(t_n, y_n) + hb^T f(et_n + ch, Y^{(m)}) \quad \text{or} \quad y_{n+1} = e_k^T Y^{(m)},$$

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 $Y^{(j)}$ can be computed in parallel.

There are several possibilities for choosing the matrix D. The most simple choice sets D=O to obtain an explicit iteration method (fixed point or functional iteration). This approach was followed in Nørsett & Simonsen [14], in Lie [12], in van der Houwen & Sommeijer [8] and in Burrage [2]. These papers deal with the iteration of implicit methods for solving nonstiff ODEs. Since we are aiming at stiff ODEs, we shall use matrices $D \ne O$. For instance, one may choose D such that we have for a prescribed number of iterations favourable stability characteristics, such as A-stability or L-stability. This approach was followed in van der Houwen et al. [9]. Alternatively, we may exploit D for improving the rate of convergence of the iteration process. For example, by identifying the diagonal elements of D with those of A we obtain the nonlinear Jacobi iteration method. In the present paper, we shall choose D such that the stability region of the iterated method rapidly converges to that of the corrector (see Subsection 3.3).

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.

Table 3.1. Starting the inner and outer iteration processes.

Order of approximation	0	1
Jacobian matrix	$\mathrm{diag}\left[J(et_n, ey_n)\right]$	$\mathrm{diag}\left[J(et_n + ch, y_n e + hcf(t_n, y_n))\right]$
Initial iterate in (3.1a)	$Y^{(0)} = y_n e,\ t^{(0)} = et_n$	$Y^{(0)} = y_n e + hcf(t_n, y_n),\ t^{(0)} = et_n + ch$
Initial Newton iterate	$Y^{(j-1)}$	$y_n e + haf(t_n, y_n) + hAf(et_n + ch, Y^{(j-1)})$

All possible combinations are equally expensive because the values of $f(t_n,y_n)$, $f(et_n+ch,Y^{(j-1)})$ 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 first-order approximations, and yields comparable accuracies.

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

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

Since each iteration step in (3.1a) essentially requires the 'wall clock time' involved in evaluating one component of $f(et_n+ch,Y^{(j-1)})$ and solving one system of dimension d, we conclude that, effectively, the work involved in performing one step by the DIRK method (3.1') consists of

$$
\left(\text{evaluation of f and J}\right) + \left(\text{LU decomposition of } I - d_j h J\right) + m\left[\text{evaluation of f} + N\left(\text{forward/backward substitution + evaluation of f}\right)\right]. \qquad (3.2)
$$

In this expression N is defined by

$$
N := \frac{N_1 + N_2 + \ldots + N_m}{m}, \qquad (3.3)
$$

with N_j denoting the number of Newton iterations for computing that component of $Y^{(j)}$ 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 [10] and [9].

Table 3.2. Summary of characteristics of DIRK, PARK and PDIRK methods of order p≥3.

Order	Stage order	Seq. stages	Processors	Stability	Reference
p=3	1	p-1	1	A-stable	Nørsett [13]
p=3	2	p-1	1	Strongly A-stable	Crouzeix [4]
p=4	1	p-1	1	A-stable	Crouzeix [4], Alexander [1]
p=4	1	p-2	2	L-stable	Iserles & Nørsett [10]
p=3, 4, 5	1	p-1	[(p+1)/2]	Strongly A-stable	van der Houwen et al. [9]
p=6, 7	1	p-1	[(p+1)/2]	Strongly A(α)-stable	ibid
p≤6, p=8	1	p	[(p+1)/2]	L-stable	ibid
p=7, 8, 10	1	p+1	[(p+1)/2]	L-stable	ibid

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

$$(3.4a)\qquad \mathbf{Y}(t_{n+1}) - \mathbf{Y}^{(j)} = [\mathbf{Y}(t_{n+1}) - \mathbf{Y}_{n+1}] + [\mathbf{Y}_{n+1} - \mathbf{Y}^{(j)}] = [\mathbf{Y}(t_{n+1}) - \mathbf{Y}_{n+1}] + Z\,[\mathbf{Y}_{n+1} - \mathbf{Y}^{(j-1)}]$$

$$= [\mathbf{Y}(t_{n+1}) - \mathbf{Y}_{n+1}] + Z^j\,[\mathbf{Y}_{n+1} - \mathbf{Y}^{(0)}],\quad j = 1, 2, \ldots ,$$

where Z is the iteration matrix defined by

$$(3.4b)\qquad Z := h\,[I - hDJ]^{-1}\,[A - D]\,J,$$

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

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

so that, after m iterations, (3.1) defines a method in which $\mathbf{Y}^{(m)}$ approximates $\mathbf{Y}(t_{n+1})$ with stage order $r^*=\min\{r,m\}$. 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.

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

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

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

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

and (3.1a) reduces to

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

We shall discuss the cases 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

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

so that the stability function is given by

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

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

$$(3.10)\qquad R_m(z) := R_{corr}(z) - z^2\mathbf{b}^TZ^m[I - zA]^{-1}\mathbf{c},$$

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

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

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

$$(3.12)\qquad R_m(z) = R_{corr}(z) + z\mathbf{b}^T[I - D^{-1}A]^mA^{-1}\mathbf{c} \quad \text{as } |z| \to \infty,$$

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

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

Since we do not know m in advance and assuming that the corrector is prescribed, this condition cannot be solved. Therefore, we conclude that diagonal iteration as defined by (3.1') is not suitable for iterating nonstiffly accurate correctors. This excludes the Gauss-Legendre formulas as suitable corrector methods (cf. Table 2.1).

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

$$(3.13)\qquad R_m(z) = R_{corr}(z) - z^{m+1}\mathbf{e}_k^T\left([I - zD]^{-1}[A - D]\right)^m[I - zA]^{-1}\mathbf{c},$$

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

$$(3.13')\qquad R_m(z) = R_{corr}(z) - [\sigma_m(z)]^m, \qquad \sigma_m(z) := \left[z^{m+1}\mathbf{e}_k^T\left([I - zD]^{-1}[A - D]\right)^m[I - zA]^{-1}\mathbf{c}\right]^{1/m}.$$

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 nonpositive 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

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

In this paper, we have tried to minimize the convergence factor at infinity, i.e., $\sigma_m(\infty)$. One possibility is to determine for each m a matrix D such that $\sigma_m(\infty)$ vanishes. However, since we do not know m in advance, this is unattractive from a computational point of view. Therefore, we determined, for a given corrector, the matrix D such that $\rho(I - D^{-1}A)$ is minimized over all possible diagonal matrices D with positive entries, and, as a posteriori test, we computed for a few values of m the 'worst' convergence factor defined by

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

Because $\sigma_m(z)$ is an analytical function in the nonpositive 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 $R_{corr}(z)$. 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. Hence, the Lobatto IIIA and the Newton-Cotes formulas seem to be less suitable as corrector methods (cf. Table 2.1). In fact, for the Newton-Cotes and Lobatto IIIA correctors we verified that (for z in the nonpositive halfplane) $\max|R_m(z)| \downarrow 1$ if $m \to \infty$, so that A-stability is only obtained in the limit. For the strongly 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 Section 4).

4. Construction of Methods

In this section, we consider a number of stiffly accurate correctors and 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 Subsection 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 to be strongly A-stable is

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

In order to save space, the correctors are presented by means of the matrix A and the vectors **a** and **c**, and the iterated versions by only giving the matrix D, because, together with the corrector, D completely defines the iterated method. Furthermore, the stage and step point orders of the methods are denoted by r and p, respectively. Finally, the range for σ_m (and the corresponding interval where the maxima are assumed), with $r\le m\le 10$ as well as the value of m_{crit} are given (cf. Section 3.3.2).

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

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

where c is a free parameter (recall that p=4 if c=1/2). An elementary calculation shows that the stability function of (4.1a) is given by

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

This function is A-acceptable for $c\ge 1/2$ and strongly A-acceptable for $c>1/2$.

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

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

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

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

leading to

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

so that the matrix D is given by

$$\text{(4.1b)} \qquad D = \frac{1}{6(1 - c)} \begin{pmatrix} \frac{c(3 - 2c)}{1 + Q} & 0 \\ 0 & \frac{2 - 3c}{1 - Q} \end{pmatrix}.$$

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=1/2 we derive from (4.1a) the Newton-Cotes corrector (with p=4 and r=3)

$$\text{(4.3a)} \qquad A = \frac{1}{24}\begin{pmatrix} 8 & -1 \\ 16 & 4 \end{pmatrix}, \quad a = \frac{1}{24}\begin{pmatrix} 5 \\ 4 \end{pmatrix}, \quad c = \begin{pmatrix} 1/2 \\ 1 \end{pmatrix}.$$

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

$$\text{(4.3b)} \qquad D = \begin{pmatrix} \frac{1}{3+\sqrt{3}} & 0 \\ 0 & \frac{1}{2(3-\sqrt{3})} \end{pmatrix}$$

with $\sigma_m \in [0.21, 0.36]$ assumed on [3.9i, 5.1i], and $m_{crit}=\infty$.

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>1/2. For example, by choosing c=3/4 we have $R(\infty)=1/3$. The corresponding Lagrange method is defined by

$$\text{(4.4a)} \qquad A = \frac{1}{288}\begin{pmatrix} 216 & -81 \\ 256 & -48 \end{pmatrix}, \quad a = \frac{1}{288}\begin{pmatrix} 81 \\ 80 \end{pmatrix}, \quad c = \begin{pmatrix} 3/4 \\ 1 \end{pmatrix}$$

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

$$\text{(4.4b)} \qquad D = \begin{pmatrix} \frac{3}{4(\sqrt{2}+1)} & 0 \\ 0 & \frac{1}{6(\sqrt{2}-1)} \end{pmatrix}$$

with $\sigma_m \in [0.21, 0.33]$ assumed on [3.2i, 4.1i] and $m_{crit}=2$.

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

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

(4.5b) $D=\frac{1}{6}\begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}$.

The Radau IIA-based method is given by:

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

(4.6b) $D=\frac{1}{30}\begin{pmatrix} 20-5\sqrt{6} & 0 \\ 0 & 12+3\sqrt{6} \end{pmatrix}$

with $\sigma_m\in[0.27, 0.35]$ assumed on [2.6i, 3.7i], and $m_{crit}=1$.

4.2. Three-processor methods

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

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

with p=r=4, and with A-acceptable stability function (see Watts & Shampine [17]). By a numerical search we found the matrix

(4.7b) $D=\begin{pmatrix} 0.12282623 & 0 & 0 \\ 0 & 0.22656820 & 0 \\ 0 & 0 & 0.32504435 \end{pmatrix}$

with $\rho(I-D^{-1}A)\approx 0.01$, $\sigma_m\in[0.49, 0.77]$ assumed on [7.1i, 8.4i], and $m_{crit}=\infty$.

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

(4.8a) $A=\frac{1}{120}\begin{pmatrix} 25-\sqrt{5} & 25-13\sqrt{5} & -1+\sqrt{5} \\ 25+13\sqrt{5} & 25+\sqrt{5} & -1-\sqrt{5} \\ 50 & 50 & 10 \end{pmatrix}$, $\mathbf{a}=\frac{1}{120}\begin{pmatrix} 11+\sqrt{5} \\ 11-\sqrt{5} \\ 10 \end{pmatrix}$, $b_0=\mathbf{e}_k^T\mathbf{a}$, $\mathbf{b}^T=\mathbf{e}_k^TA$, $\mathbf{c}=\frac{1}{10}\begin{pmatrix} 5-\sqrt{5} \\ 5+\sqrt{5} \\ 10 \end{pmatrix}$,

with p=6 and r=4, and with A-acceptable stability function (see Dekker and Verwer [5]). The iterated version is generated by

(4.8b) $D=\begin{pmatrix} 0.48015157 & 0 & 0 \\ 0 & 0.10941809 & 0 \\ 0 & 0 & 0.16037253 \end{pmatrix}$

with $\rho(I-D^{-1}A)\approx 0.0043$, $\sigma_m\in[0.52, 0.88]$ assumed on [8.9i, 10i], and $m_{crit}=\infty$.

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

(4.9a) $A=\frac{1}{120960}\begin{pmatrix} 98392 & -81634 & 31213 \\ 112000 & -61600 & 28000 \\ 110592 & -48384 & 36288 \end{pmatrix}$, $\mathbf{a}=\frac{1}{120960}\begin{pmatrix} 22589 \\ 22400 \\ 22464 \end{pmatrix}$, $b_0=\mathbf{e}_k^T\mathbf{a}$, $\mathbf{b}^T=\mathbf{e}_k^TA$, $\mathbf{c}=\frac{1}{12}\begin{pmatrix} 7 \\ 10 \\ 12 \end{pmatrix}$

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

$$(4.9b)\qquad D = \begin{pmatrix} 0.21051645 & 0 & 0 \\ 0 & 0.28849216 & 0 \\ 0 & 0 & 0.33912361 \end{pmatrix}$$

with $\rho(I-D^{-1}A)=0.011$, $\sigma_m \in [0.49, 0.69]$ assumed on [5.1i, 6.2i] and $m_{crit}=3$.

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

$$(4.10a)\qquad A := \begin{pmatrix} \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{pmatrix},\quad \mathbf{a = 0},\ b_0 = e_k{}^T a,\ \mathbf{b}^T = e_k{}^T A,\ \mathbf{c = Ae}$$

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

$$(4.10b)\qquad D = \begin{pmatrix} 0.32039049 & 0 & 0 \\ 0 & 0.13997017 & 0 \\ 0 & 0 & 0.37167618 \end{pmatrix}$$

with $\rho(I-D^{-1}A)\approx 0.0047$, $\sigma_m \in [0.52, 1.0]$ assumed on [6.6i, 9.3i] and $m_{crit}=5$.

4.3. Four-processor methods

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

$$(4.11a)\qquad A = \frac{1}{2880}\begin{pmatrix} 646 & -264 & 106 & -19 \\ 992 & 192 & 32 & -8 \\ 918 & 648 & 378 & -27 \\ 1024 & 384 & 1024 & 224 \end{pmatrix},\quad a = \frac{1}{2880}\begin{pmatrix} 251 \\ 232 \\ 243 \\ 224 \end{pmatrix},\quad b_0 = e_k{}^T a,\ \mathbf{b}^T = e_k{}^T A,\ c = \frac{1}{4}\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

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

$$(4.11b)\qquad D = \begin{pmatrix} 0.09220187 & 0 & 0 & 0 \\ 0 & 0.16837302 & 0 & 0 \\ 0 & 0 & 0.24013828 & 0 \\ 0 & 0 & 0 & 0.30942512 \end{pmatrix}$$

with $\rho(I-D^{-1}A)\approx 0.1$, $\sigma_m \in [0.76, 1.04]$ assumed on [8.7i, 11.8i] and $m_{crit}=\infty$.

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

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

(4.12a)

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

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

$$\text{(4.12b)} \quad D = \begin{pmatrix} 0.29809917 & 0 & 0 & 0 \\ 0 & 0.18143990 & 0 & 0 \\ 0 & 0 & 0.06466021 & 0 \\ 0 & 0 & 0 & 0.16221872 \end{pmatrix}$$

with $\rho(I-D^{-1}A)\approx 0.021$, $\sigma_m \in [0.87, 1.32]$ assumed on [15.4i, 19i], and $m_{crit}=\infty$.

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

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

$$\text{(4.13a)} \quad \mathbf{a} = \frac{1}{332640}\begin{pmatrix} 22436 \\ 6811 \\ 10043 \\ 9936 \end{pmatrix}, \quad b_0 = \mathbf{e_k}^T\mathbf{a}, \quad \mathbf{b}^T = \mathbf{e_k}^T A, \quad \mathbf{c} = \frac{1}{12}\begin{pmatrix} 2 \\ 7 \\ 11 \\ 12 \end{pmatrix}.$$

The iterated method is generated by

$$\text{(4.13b)} \quad D = \begin{pmatrix} 0.13380299 & 0 & 0 & 0 \\ 0 & 0.11358038 & 0 & 0 \\ 0 & 0 & 0.22689850 & 0 \\ 0 & 0 & 0 & 0.25010131 \end{pmatrix}$$

with $\rho(I-D^{-1}A)\approx 0.045$, $\sigma_m \in [0.59, 0.93]$ assumed on [8.2i, 11.8i], and $m_{crit}=6$.

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

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

$$\text{(4.14a)} \quad \mathbf{a} = \mathbf{0}, \quad b_0 = 0, \quad \mathbf{b}^T = \mathbf{e_k}^T A, \quad \mathbf{c} = A\mathbf{e}$$

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

$$\text{(4.14b)} \quad D = \begin{pmatrix} 0.32049937 & 0 & 0 & 0 \\ 0 & 0.08915379 & 0 & 0 \\ 0 & 0 & 0.18173957 & 0 \\ 0 & 0 & 0 & 0.23336280 \end{pmatrix}$$

with $\rho(I-D^{-1}A)\approx 0.024$, $\sigma_m \in [0.74, 1.31]$ assumed on [10.0i, 17.2i], and $m_{crit}=7$.

4.4. Survey of methods

In Table 4.1, we have summarized a few characteristics of the methods derived in the preceding subsections.

Table 4.1. Main characteristics of diagonally iterated IRK methods.

Method	p	r	k	$\rho(I-D^{-1}A)$	σ_m-range ($r\le m\le 10$)	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

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 Section 5.4.

5. Numerical Experiments

In this section, the 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 Section 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 Section 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 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 step length, $L:=T-t_0$ 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 [15]

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

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

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

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

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

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

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

The third test problem is that of Kaps [11]:

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

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 et al. [7]).

The test set of Enright et al. [6] contains the following system of ODEs describing a chemical reaction:

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

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

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

At t=T=51 we found the approximate solution

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

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

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

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 mesh size 1/40 leads to a system of 39 ODEs whose exact solution is given by $(j/40)^2\cos(t)$, j=1,...,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 A.1a and A.1b in the Appendix to this paper present Δ values for the various test problems obtained for L/h=1, 2, 4, 8, 16. From these results we can derive for each test problem the effective orders by computing (Δ(h)-Δ(2h))/0.3. For h we chose the smallest value for which results are available. The resulting effective orders are listed in Table 5.1. For each problem, the result of the most accurate corrector is indicated in bold.

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.

Table 5.1. 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

5.3. Performance of the iteration process for two-processor correctors.

In this subsection, 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 results for both correctors. Furthermore, in the Tables 5.2 and 5.3 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 Tables A.2 until A.5 of the Appendix. In the tables of results, 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). We list for several values of L/h the accuracies corresponding to the correctors of Lagrange type (first column), of Gauss-Legendre (second column), and of Radau IIA (third column). These results confirm that, in general, the Gauss corrector is not suited to be iterated by diagonal iteration methods.

Table 5.2. 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	3.8	4.1	-2.2	5.3	4.0	*	4.8	3.6	*	5.0	2.7	*	5.3
2	**5.0**	1.0	**4.2**	**5.8**	-1.1	**4.7**	6.5	*	**5.2**	6.7	*	5.9	6.7	*	6.7
3		**1.9**			2.4		**6.7**	2.9		**7.7**	3.9	**5.8**	8.4	1.9	**6.4**
4					**2.5**			**3.1**			**3.8**		**8.7**	**4.6**	

Table 5.3. 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	1.7	2.3	1.5	2.1	2.6	*	2.4	2.8	*	2.7	3.1	*	3.0
2	3.4	2.6	2.9	3.9	2.9	3.5	4.5	*	4.1	5.2	*	4.7	5.8	*	5.3
3	4.3	3.8	3.6	5.4	4.8	4.5	6.4	5.7	5.4	7.4	6.6	6.3	8.3	6.9	7.2
4	**4.5**	4.7	**3.4**	**5.7**	5.9	**4.3**	**6.9**	7.1	**5.2**	8.1	8.3	**6.1**	9.3	9.5	**7.0**
5		**5.0**			**6.1**			**7.3**		**8.2**	**8.5**		**9.4**	**9.7**	

5.4. Efficiency of diagonally iterated IRK correctors

In this final subsection, we compare the efficiency of the diagonally iterated IRK correctors with three fourth-order DIRK methods from the literature, viz. the three-stage method generated by the Butcher array

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

(cf. Crouzeix [4] and Alexander [1]), and the four-stage, parallel DIRK methods of Iserles and Nørsett [10]:

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

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

The method (5.6) is A-stable and requires three sequential stages per step. The methods (5.7) and (5.8) are A-stabele 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 5.4 and 5.5, 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' integration results, are indicated in bold. As in the preceding subsection, we only present results for the problems (5.2) and (5.4). Results for the additional test problems may be found in the Appendix.

In the case of the nonlinear Prothero-Robinson problem, Table 5.4a 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 2 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 $Y^{(0)}$ 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 less good 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 5.4a 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 5.4b 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 5.5a 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 5.4b is given by Table 5.5b. It reveals that the iterated methods are at least competitive with the parallel DIRK methods, but usually much more efficient.

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.

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 dissuaded.

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 Lagrange corrector has a slightly larger stage order than 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 'conventinal' 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 remarkebly high accuracies.

Table 5.4a. 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	...	m=∞
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)	2		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 5.4b. 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

Table 5.5a. 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	...	m=∞
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		**2.1**	3.4	4.3	4.5								4.5
Lagrange (4.4a			**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		**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		**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

Table 5.5b. 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**

ACKNOWLEDGEMENT

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

REFERENCES

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

[2] **Burrage, K. [1989]**: The error behaviour of a general class of predictor-corrector methods, CMSR Report, University of Liverpool.

[3] **Butcher, J.C. [1987]**: The numerical analysis of ordinary differential equations, Runge-Kutta and general linear methods, Wiley, New York.

[4] **Crouzeix, M. [1975]**: Sur l'approximation des équations différentielles opérationnelles linéaires par des méthodes de Runge-Kutta, Ph. D. Thesis, Université de Paris.

[5] **Dekker, K. & Verwer, J.G. [1984]**: Stability of Runge-Kutta methods for stiff nonlinear differential equations, CWI Monograph 2, North-Holland, Amsterdam.

[6] **Enright, W.H., Hull, T.E. & Lindberg, B. [1975]**: Comparing numerical methods for stiff systems of ODEs, BIT **15**, 10-48.

[7] **Hairer, E., Lubich, Ch. & Roche, M. [1988]**: Error of Runge-Kutta methods for stiff problems studied via differential algebraic equations, BIT **28**, 678-700.

[8] **Houwen, P.J. van der & Sommeijer, B.P. [1990]**: Parallel iteration of high-order Runge-Kutta methods with stepsize control, J. Comp. Appl. Math. **29**, 111-127.

[9] **Houwen, P.J. van der, Sommeijer, B.P. & Couzy, W. [1989]**: Embedded diagonally implicit Runge-Kutta algorithms on parallel computers, Report NM-R8912, Centre for Mathematics and Computer Science, Amsterdam (submitted for publication).

[10] **Iserles, A. & Nørsett, S.P. [1988]**: On the theory of parallel Runge-Kutta methods, Report DAMTP 1988/NA12, University of Cambridge.

[11] **Kaps, P. [1981]**: 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.

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

[13] **Nørsett, S.P. [1974]**: Semi-explicit Runge-Kutta methods, Report Mathematics and Computation No.6/74, Depart. of Mathematics, University of Trondheim.

[14] **Nørsett, S.P. & Simonsen, H.H. [1989]**: Aspects of parallel Runge-Kutta methods, in: Numerical methods for ordinary differential equations, A. Bellen, C.W. Gear & E. Russo (eds.), Proceedings L'Aquila 1987, Lecture Notes in Mathematics **1386**, Springer-Verlag, Berlin.

[15] **Prothero, A. & Robinson, A. [1974]**: On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations, Math. Comp. **28**, 145-162.

[16] **Sommeijer, B.P., Couzy, W. & van der Houwen, P.J. [1989]**: A-stable parallel block methods, Report NM-R8918, Centre for Mathematics and Computer Science, Amsterdam (submitted for publication).

[17] **Watts, H.A. & Shampine, L.F. [1972]**: A-stable block implicit one-step methods, BIT **12**, 252-266.

Appendix

This Appendix lists a large number of experiments which give additional support to the theory developed above.

A.1. Effective orders of the correctors

The Tables A.1a and A.1b present Δ-values for the various test problems obtained for L/h=1, 2, 4, 8, 16 by iterating the corrector to convergence. The effective orders in Table 5.1 are derived from these tables.

Table A.1a. 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 A.1b. 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	-	-

A.2. Performance of the iteration process for two-processor correctors

Here we give the results for the two-processor correctors applied to the remaining test problems (5.1), (5.3) and (5.5). Similar to the procedure described in Section 5.3, we give results for the most accurate one of the Lagrange-type correctors (4.3a) and (4.4a), the Gauss-Legendre corrector and the Radau IIA corrector. The following tables contain the accuracies obtained for several values of L/h. Again, the 1st, 2nd and 3rd column for each L/h-value corresponds to the Lagrange, Gauss and Radau corrector, respectively.

Table A.2a. Values of Δ for Problem (5.1) 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	3.9	4.2	-1.8	5.3	4.0	*	4.9	3.7	*	5.1	2.9	*	5.3
2	**5.1**	1.0	**4.2**	5.8	-0.8	**4.7**	6.4	*	**5.3**	6.6	*	**5.9**	6.7	*	6.8
3		**1.9**		**5.9**	**2.5**		**6.8**	3.0		7.7	3.0		8.3	2.1	**6.5**
4								**3.1**		**7.8**	**3.8**		**8.8**	**4.7**	

Table A.2b. Values of Δ for Problem (5.3) with $\varepsilon=10^{-3}$ obtained by iterating the Newton-Cotes corrector (4.3a), 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	1.1	-0.1	1.6	1.5	-2.1	1.7	1.9	-6.0	2.0	2.3	*	2.2	2.5	*	2.5
2	2.3	1.3	3.0	2.9	-0.8	3.0	3.6	-4.0	3.3	4.2	*	3.8	4.8	*	4.4
3	3.6	1.3	**2.4**	4.8	1.7	3.4	5.4	2.0	4.3	6.2	2.2	5.3	**7.0**	2.7	6.1
4	**3.3**	**1.2**		**4.3**	**1.8**	**3.2**	**5.1**	**2.4**	**4.1**	**5.9**	3.1	**5.0**		4.2	**5.9**
5											**3.2**			**4.3**	

Table A.2c. Values of Δ for Problem (5.3) with $\varepsilon=10^{-8}$ obtained by iterating the Newton-Cotes corrector (4.3a), 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	1.1	-0.1	1.6	1.5	-7.0	1.7	1.9	*	2.0	2.3	*	2.2	2.6	*	2.5
2	2.3	1.3	3.0	2.9	-6.0	3.0	3.6	*	3.3	4.2	*	3.8	4.8	*	4.4
3	3.7	1.3	**2.4**	4.8	1.7	3.4	5.4	1.9	4.3	6.2	1.7	5.3	7.0	0.4	6.2
4	**3.3**	**1.2**		**4.5**	**1.8**	3.3	**5.7**	**2.4**	**4.1**	**6.9**	**3.0**	**5.0**	**8.1**	**3.6**	**5.9**
5						**3.2**									

Table A.2d. Values of Δ for Problem (5.5) obtained by iterating the Newton-Cotes corrector (4.3a), 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	1.5	0.4	1.8	2.0	-2.0	2.1	2.4	*	2.3	2.7	*	2.6	2.8	*	2.9
2	2.9	1.1	**2.5**	3.4	-0.9	3.4	3.9	*	4.1	4.4	*	4.2	4.9	*	4.6
3	**3.2**	2.0		**4.2**	**2.6**	**3.2**	5.0	**3.2**	4.1	5.6	3.2	5.0	6.2	2.1	6.2
4		**1.9**					5.3		**4.0**	6.4	**3.9**	4.9	7.3	**4.8**	**5.7**
5							**5.4**			**6.5**		**4.8**	**7.7**		

Also for these problems we observe that the iterated methods based on the Gauss-Legendre corrector, which is not stiffly accurate, are less robust than the Lagrange and Radau based methods. Therefore, we omit the Gauss method in the tests on the efficiency of the various DIRK methods.

A.3. Efficiency of diagonally iterated IRK methods

We conclude with a comparison of the efficiency of the diagonally iterated correctors with high stage order and the DIRK methods (5.6), (5.7) and (5.8). Therefore we applied these methods to the remaining test problems (5.1), (5.3) and (5.5) and list the results in the following tables. We use the same test procedure and notation as described in Section 5.4.

For all these problems the 'genuine' DIRK methods show a second-order behaviour (i.e., their stage order+1) and, consequently, are markedly less efficient than the diagonally iterated methods which have much higher effective orders (cf. Table 5.1).

Table A.3a. Problem (5.1): 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	...	m=∞
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)	2		3.5	4.7										4.7
Lagrange (4.4)			3.5	**5.1**										5.1
Radau IIA (4.6)			**3.9**	4.2										4.2
Newton-C. (4.7)	3		3.2	3.8	**5.8**	6.1								6.1
Lobatto IIIA (4.8)			3.0	2.8	5.0	6.1								6.1
Lagrange (4.9)			3.3	4.0	5.6	**6.5**								6.5
Radau IIA (4.10)			3.4	3.1	5.2	5.0								5.0
Newton-C. (4.11)	4		3.2	3.7	5.0	4.7	5.3	6.0	7.1	7.1	7.0			7.0
Lobatto IIIA (4.12)			2.7	2.2	2.4	4.0	4.7	5.7	7.0	7.0	7.1			7.1
Lagrange (4.13)			3.0	2.9	3.1	3.9	5.0	6.1	**7.8**	**7.5**				**7.5**
Radau IIA (4.14)			3.0	2.9	3.1	4.6	5.6	6.3						6.3
Crouzeix-Alex. (5.6)	1	2	-	-	2.4									-
Iserles-Nørsett (5.7)	2		-	2.4										-
Iserles-Nørsett (5.8)			-	2.7										-
Newton-C. (4.3)	2		4.1	5.3	5.4									5.4
Lagrange (4.4)			4.2	**5.8**	5.9									5.9
Radau IIA (4.6)			**5.3**	4.7										4.7
Newton-C. (4.7)	3		3.5	3.6	**7.0**	7.3								7.3
Lobatto IIIA (4.8)			3.0	2.2	6.7	6.6	7.3							7.3
Lagrange (4.9)			3.6	3.8	5.9	**7.9**	**7.7**	**7.6**						7.6
Radau IIA (4.10)			3.9	2.8	5.7	5.8	6.0							6.0
Newton-C. (4.11)	4		3.4	3.3	5.4	5.2	5.3	6.0	6.8	7.7	8.5	8.2		8.2
Lobatto IIIA (4.12)			2.4	1.3	1.5	4.6	4.5	5.6	6.7	7.3	8.4			8.4
Lagrange (4.13)			3.0	2.3	2.8	5.0	5.2	6.3	7.5	**8.9**	**9.0**	**8.9**		**8.9**
Radau IIA (4.14)			2.9	2.3	2.7	5.0	5.9	7.0	7.5	7.4				7.4
Crouzeix-Alex. (5.6)	1	4	-	-	2.7									-
Iserles-Nørsett (5.7)	2		-	2.9										-
Iserles-Nørsett (5.8)			-	3.2										-
Newton-C. (4.3)	2		3.9	5.7	6.0									6.0
Lagrange (4.4)			4.0	**6.4**	6.8									6.8
Radau IIA (4.6)			**4.9**	5.3										5.3
Newton-C. (4.7)	3		3.1	3.1	**8.3**	**8.3**	8.5	8.5						8.5
Lobatto IIIA (4.8)			2.4	1.0	6.7	6.9	8.4	8.6						8.6
Lagrange (4.9)			3.3	3.6	6.1	8.2	**9.8**	**8.8**						8.8
Radau IIA (4.10)			3.7	2.1	5.6	6.4	6.9							6.9
Newton-C. (4.11)	4		3.0	2.6	5.4	5.6	5.6	6.1	6.8	7.7	8.7	10.0		9.5
Lobatto IIIA (4.12)			1.3	-1.0	-0.5	4.4	4.2	6.0	6.1	7.3	8.0	8.9		9.6
Lagrange (4.13)			2.3	1.0	1.7	5.2	5.6	6.6	7.8	**8.8**	**9.8**	**10.9**		**10.5**
Radau IIA (4.14)			2.2	0.9	1.5	5.2	6.2	7.1	8.7	8.7	8.6			8.6

Table A.3b. Problem (5.1): 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	2.9	3.6
Iserles-Nørsett (5.8)	4	2	2	2.7	3.2	3.8
Newton-C. (4.3)	4	4	2	4.7	5.4	6.0
Newton-C. (4.7)	4	4	3	6.1	7.3	**8.3**
Lobatto IIIA (4.8)	4	4	3	6.1	6.6	6.9
Lagrange (4.9)	4	4	3	**6.5**	**7.9**	8.2
Radau IIA (4.10)	4	4	3	5.0	5.8	6.4
Newton-C. (4.11)	4	4	4	4.7	5.2	5.6
Lobatto IIIA (4.12)	4	4	4	4.0	4.6	4.4
Lagrange (4.13)	4	4	4	3.9	5.0	5.2
Radau IIA (4.14)	4	4	4	4.6	5.0	5.2

Table A.4a. Problem (5.3), $\varepsilon=10^{-3}$: Results for diagonally iterated correctors and for (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	...	m=∞
Crouzeix-Alex. (5.6)	1	1	-	-	0.8									-
Iserles-Nørsett (5.7)	2		-	1.1										-
Iserles-Nørsett (5.8)			-	1.2										-
Newton-C. (4.3)	2		1.1	2.3	3.6	3.3								3.3
Lagrange (4.4)			1.1	2.1	2.6	2.7								2.7
Radau IIA (4.6)			**1.6**	3.0	2.4									2.4
Newton-C. (4.7)	3		0.9	2.8	3.4	**4.5**	4.2							4.2
Lobatto IIIA (4.8)			0.8	1.9	2.7	3.8	4.9	4.7						4.7
Lagrange (4.9)			0.9	**3.2**	**3.6**	4.1	3.9	3.8	3.8					3.8
Radau IIA (4.10)			1.0	2.1	2.7	3.6	4.0							4.0
Newton-C. (4.11)	4		0.9	2.5	3.3	4.3	**4.8**	5.8	5.5	5.4				5.4
Lobatto IIIA (4.12)			0.6	1.4	2.1	3.0	4.2	5.2	5.6					5.6
Lagrange (4.13)			0.8	1.8	2.8	3.7	**4.8**	**6.2**	**5.9**	**5.8**				**5.8**
Radau IIA (4.14)			0.8	1.8	2.9	4.1	**4.8**	5.1	5.0					5.0
Crouzeix-Alex. (5.6)	1	2	-	-	2.0									-
Iserles-Nørsett (5.7)	2		-	2.9										-
Iserles-Nørsett (5.8)			-	2.2										-
Newton-C. (4.3)	2		1.5	2.9	4.8	4.3								4.3
Lagrange (4.4)			1.6	2.8	3.5	3.5	3.6							3.6
Radau IIA (4.6)			**1.7**	3.0	3.4	3.2								3.2
Newton-C. (4.7)	3		1.4	3.3	4.5	**5.9**	5.4							5.4
Lobatto IIIA (4.8)			1.2	1.9	3.6	4.9	6.0							6.0
Lagrange (4.9)			1.4	**3.4**	**4.9**	5.1	5.0							5.0
Radau IIA (4.10)			1.5	2.4	3.7	4.8	5.2	5.2	5.3					5.3
Newton-C. (4.11)	4		1.3	2.9	4.3	5.6	5.1	5.7	6.8	6.8	6.7			6.7
Lobatto IIIA (4.12)			1.1	0.9	1.1	3.4	4.2	5.2	7.4	6.6	6.8			6.8
Lagrange (4.13)			1.2	1.9	2.4	4.6	5.4	6.6	**7.4**	**7.2**				**7.2**
Radau IIA (4.14)			1.2	1.9	2.3	5.5	**5.7**	**6.8**	6.4					6.4
Crouzeix-Alex. (5.6)	1	4	-	-	3.9									-
Iserles-Nørsett (5.7)	2		-	3.1										-
Iserles-Nørsett (5.8)			-	3.1										-
Newton-C. (4.3)	2		1.9	**3.6**	5.4	5.1								5.1
Lagrange (4.4)			**2.0**	3.5	4.3	4.4								4.4
Radau IIA (4.6)			**2.0**	3.3	4.3	4.1								4.1
Newton-C. (4.7)	3		1.7	2.8	5.6	**7.0**	6.6							6.6
Lobatto IIIA (4.8)			1.6	0.7	4.6	5.7	**7.5**	7.3						7.3
Lagrange (4.9)			1.8	3.5	**5.8**	6.2	6.1							6.1
Radau IIA (4.10)			1.9	1.9	4.6	5.9	6.3							6.3
Newton-C. (4.11)	4		1.7	2.4	5.4	5.9	5.6	6.0	6.7	7.7	8.2	8.0		8.0
Lobatto IIIA (4.12)			1.0	-1.2	-0.9	5.2	4.0	5.5	6.1	7.0	7.8	8.3		8.2
Lagrange (4.13)			1.6	0.7	1.4	5.3	5.9	6.9	**8.1**	**8.8**				**8.8**
Radau IIA (4.14)			1.6	0.6	1.1	5.6	6.2	**7.4**	7.7	7.8				7.8

Table A.4b. Problem (5.3), $\varepsilon=10^{-3}$: 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	2.9	3.1	3.7
Iserles-Nørsett (5.8)	4	2	2	2.2	3.1	3.9
Newton-C. (4.3)	4	4	2	3.3	4.3	5.1
Newton-C. (4.7)	4	4	3	**4.5**	**5.9**	**7.0**
Lobatto IIIA (4.8)	4	4	3	3.8	4.9	5.7
Lagrange (4.9)	4	4	3	4.1	5.1	6.2
Radau IIA (4.10)	4	4	3	3.6	4.8	5.9
Newton-C. (4.11)	4	4	4	4.3	5.6	5.9
Lobatto IIIA (4.12)	4	4	4	3.0	3.4	5.2
Lagrange (4.13)	4	4	4	3.7	4.6	5.3
Radau IIA (4.14)	4	4	4	4.1	5.5	5.6

Table A.5a. Problem (5.3), $\varepsilon=10^{-8}$: Results for diagonally iterated correctors and for (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	...	m=∞
Crouzeix-Alex. (5.6)	1	1	-	-	0.8									-
Iserles-Nørsett (5.7)	2		-	1.1										-
Iserles-Nørsett (5.8)			-	1.2										-
Newton-C. (4.3)	2		1.1	2.3	3.7	3.3								3.3
Lagrange (4.4)			1.1	2.1	2.6	2.7								2.7
Radau IIA (4.6)			**1.6**	3.0	2.4									2.4
Newton-C. (4.7)	3		0.9	2.9	3.4	**4.9**	4.2							4.2
Lobatto IIIA (4.8)			0.8	1.9	2.7	3.8	5.2	5.4						5.4
Lagrange (4.9)			0.9	**3.2**	3.6	4.1	3.9							3.9
Radau IIA (4.10)			1.0	2.1	2.7	3.6	4.2	4.3	4.4					4.4
Newton-C. (4.11)	4		0.9	2.5	3.3	4.3	**5.3**	5.8	5.9					5.9
Lobatto IIIA (4.12)			0.6	1.5	3.0	3.9	4.6	5.6	6.7	9.8	7.8			7.8
Lagrange (4.13)			0.8	1.8	**4.2**	4.2	5.1	**6.9**	6.1	6.0				6.0
Radau IIA (4.14)			0.8	1.8	3.3	4.2	4.8	5.6	**6.3**	**6.6**				**6.6**
Crouzeix-Alex. (5.6)	1	2	-	-	2.0									-
Iserles-Nørsett (5.7)	2		-	2.9										-
Iserles-Nørsett (5.8)			-	2.2										-
Newton-C. (4.3)	2		1.5	2.9	4.8	4.5								4.5
Lagrange (4.4)			1.6	2.8	3.5	3.6								3.6
Radau IIA (4.6)			**1.7**	3.0	3.4	3.3	3.2							3.2
Newton-C. (4.7)	3		1.4	**4.7**	4.5	**5.9**	5.5	5.4						5.4
Lobatto IIIA (4.8)			1.2	2.6	3.6	4.9	6.4	7.3	7.2					7.2
Lagrange (4.9)			1.4	3.4	**4.9**	5.1	5.0							5.0
Radau IIA (4.10)			1.5	2.8	3.7	4.8	5.6	5.8						5.8
Newton-C. (4.11)	4		1.3	3.5	4.3	5.6	**6.8**	7.6	7.7					7.7
Lobatto IIIA (4.12)			1.1	2.3	3.9	5.3	6.2	7.4	8.7	10.4	10.3	10.2		10.2
Lagrange (4.13)			1.2	2.5	**4.9**	5.5	6.6	**7.7**	7.4					7.4
Radau IIA (4.14)			1.2	2.5	4.2	5.7	6.4	7.4	**8.3**	**8.7**				**8.7**
Crouzeix-Alex. (5.6)	1	4	-	-	3.8									-
Iserles-Nørsett (5.7)	2		-	3.1										-
Iserles-Nørsett (5.8)			-	3.1										-
Newton-C. (4.3)	2		1.9	3.6	5.4	5.7								5.7
Lagrange (4.4)			**2.0**	3.5	4.3	4.4								4.4
Radau IIA (4.6)			**2.0**	3.3	4.3	4.1								4.1
Newton-C. (4.7)	3		1.7	**4.4**	5.6	**7.0**	6.7							6.7
Lobatto IIIA (4.8)			1.6	3.3	4.6	6.1	7.8	9.2	9.0					9.0
Lagrange (4.9)			1.8	3.9	**6.2**									6.2
Radau IIA (4.10)			1.9	3.5	4.6	5.9	7.1	7.3						7.3
Newton-C. (4.11)	4		1.7	**4.4**	5.4	6.8	**8.3**	**9.3**	9.5	9.6				9.6
Lobatto IIIA (4.12)			1.5	2.9	3.8	6.7	7.7	9.2	**10.7**	**12.6**				**12.6**
Lagrange (4.13)			1.6	3.2	5.7	6.7	8.1	8.9	8.8					8.8
Radau IIA (4.14)			1.6	3.2	5.1	**7.0**	7.9	9.2	10.4	10.8				10.8

Table A.5b. Problem (5.3), $\varepsilon=10^{-8}$: 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	2.9	3.1	3.7
Iserles-Nørsett (5.8)	4	2	2	2.2	3.1	3.8
Newton-C. (4.3)	4	4	2	3.3	4.5	5.7
Newton-C. (4.7)	4	4	3	**4.9**	**5.9**	**7.0**
Lobatto IIIA (4.8)	4	4	3	3.8	4.9	6.1
Lagrange (4.9)	4	4	3	4.1	5.1	6.2
Radau IIA (4.10)	4	4	3	3.6	4.8	5.9
Newton-C. (4.11)	4	4	4	4.3	5.6	6.8
Lobatto IIIA (4.11)	4	4	3	3.9	5.3	6.7
Lagrange (4.13)	4	4	4	4.2	5.5	6.7
Radau IIA (4.14)	4	4	4	4.2	5.7	**7.0**

Table A.6a. Problem (5.5): 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	...	m=∞
Crouzeix-Alex. (5.6)	1	1	-	-	*									-
Iserles-Nørsett (5.7)	2		-	*										-
Iserles-Nørsett (5.8)			-	*										-
Newton-C. (4.3)	2		1.5	2.9	**3.2**									3.2
Lagrange (4.4)			1.6	**3.2**	3.1									3.1
Radau IIA (4.6)			**1.8**	2.5										2.5
Newton-C. (4.7)	3		1.4	2.4	3.1	**4.5**	**4.9**	**4.6**						4.6
Lobatto IIIA (4.8)			1.2	1.4	2.3	3.4	3.7	5.0	4.7	4.8				4.8
Lagrange (4.9)			1.4	2.6	3.1	4.4	4.5							4.5
Radau IIA (4.10)			1.5	1.7	2.5	3.4	3.5	3.6						3.6
Newton-C. (4.11)	4		1.3	2.2	3.1	3.5	3.8	4.3	**5.4**	5.5	5.6	5.7		5.7
Lobatto IIIA (4.12)			1.0	1.0	1.6	1.9	2.8	3.3	3.8	4.7	5.2	5.9		6.0
Lagrange (4.13)			1.2	1.5	2.0	2.6	3.7	4.4	4.9	**5.9**	**6.3**	**6.4**		**6.4**
Radau IIA (4.14)			1.2	1.5	2.1	2.6	3.7	**4.6**	5.1	5.4	5.2			5.2
Crouzeix-Alex. (5.6)	1	2	-	-	*									-
Iserles-Nørsett (5.7)	2		-	2.4										-
Iserles-Nørsett (5.8)			-	2.7										-
Newton-C. (4.3)	2		2.0	3.4	**4.2**									4.2
Lagrange (4.4)			**2.1**	**3.9**	4.0									4.0
Radau IIA (4.6)			**2.1**	3.4	3.2									3.2
Newton-C. (4.7)	3		1.9	2.7	3.7	**4.9**	**5.8**	**5.9**						5.9
Lobatto IIIA (4.8)			1.8	1.4	2.8	4.2	4.6	5.3	6.1	6.1	6.2			6.2
Lagrange (4.9)			1.9	2.9	3.7	**4.9**	5.5	5.6						5.6
Radau IIA (4.10)			2.0	1.9	2.9	4.2	4.5	4.7	4.8					4.8
Newton-C. (4.11)	4		1.8	2.5	3.7	4.4	4.4	4.9	5.6	6.5	7.3	7.2		7.4
Lobatto IIIA (4.12)			1.4	*	1.1	2.8	3.4	4.0	4.6	5.3	6.0	6.6		7.7
Lagrange (4.13)			1.7	1.5	2.4	3.6	4.2	5.5	**6.3**	**7.0**	**7.8**			**7.8**
Radau IIA (4.14)			1.7	1.5	2.3	3.7	4.1	4.9	5.7	6.4	6.7	6.6		6.5
Crouzeix-Alex. (5.6)	1	4	-	-	2.8									-
Iserles-Nørsett (5.7)	2		-	3.0										-
Iserles-Nørsett (5.8)			-	3.3										-
Newton-C. (4.3)	2		**2.4**	3.9	**5.0**	**5.3**	5.4							5.4
Lagrange (4.4)			**2.4**	4.0	**5.0**	4.9	4.8							4.8
Radau IIA (4.6)			2.3	**4.1**	4.1	4.0								4.0
Newton-C. (4.7)	3		2.2	2.8	4.3	**5.3**	**6.4**	**7.0**	**7.2**					7.2
Lobatto IIIA (4.8)			1.9	*	3.4	4.6	5.4	5.9	7.0	7.4	7.6	7.6		7.7
Lagrange (4.9)			2.2	3.2	4.4	**5.3**	6.3	6.7	6.7	6.8				6.8
Radau IIA (4.10)			2.3	1.8	3.5	4.6	5.6	5.8	6.0	6.1				6.1
Newton-C. (4.11)	4		2.1	2.4	4.3	5.1	5.1	5.5	6.1	6.8	7.7	8.5		9.2
Lobatto IIIA (4.12)			*	*	*	3.0	4.0	4.6	5.3	6.0	6.7	7.3		**9.5**
Lagrange (4.13)			1.8	*	1.7	4.2	4.8	5.6	6.6	**7.4**	**8.2**	**9.0**		9.3
Radau IIA (4.14)			1.6	*	1.5	4.2	4.7	5.5	6.3	7.1	8.0			8.0

Table A.6b. Problem (5.5): 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	3.3	3.9
Newton-C. (4.3)	4	4	2	3.2	4.2	**5.3**
Newton-C. (4.7)	4	4	3	**4.5**	**4.9**	**5.3**
Lobatto IIIA (4.8)	4	4	3	3.4	4.2	4.6
Lagrange (4.9)	4	4	3	4.4	**4.9**	**5.3**
Radau IIA (4.10)	4	4	3	3.4	4.2	4.6
Newton-C. (4.11)	4	4	4	3.5	4.4	5.1
Lobatto IIIA (4.12)	4	4	4	1.9	2.8	3.0
Lagrange (4.13)	4	4	4	2.6	3.6	4.2
Radau IIA (4.14)	4	4	4	2.6	3.7	4.2