ELSEVIER

Applied Numerical Mathematics 25 (1997) 257–274

APPLIED NUMERICAL MATHEMATICS

The solution of implicit differential equations on parallel computers ☆

P.J. van der Houwen *, W.A. van der Veen

CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract

We construct and analyze parallel iterative solvers for the solution of the linear systems arising in the application of Newton's method to s-stage implicit Runge–Kutta (RK) type discretizations of implicit differential equations (IDEs). These linear solvers are partly iterative and partly direct. Each linear system iteration again requires the solution of linear subsystems, but now only of IDE dimension, which is s times less than the dimension of the linear system in Newton's method. Thus, the effective costs on a parallel computer system are only one LU-decomposition of IDE dimension for each Jacobian update, yielding a considerable reduction of the effective LU-costs. The method parameters can be chosen such that only a few iterations by the linear solver are needed. The algorithmic properties are illustrated by solving the transistor problem (index 1) and the car axis problem (index 3) taken from the CWI test set. © 1997 Elsevier Science B.V.

Keywords: Numerical analysis; Implicit differential equations; DAEs; Runge–Kutta methods; Parallelism

1. Introduction

We consider initial value problems (IVPs) for systems of implicit differential equations (IDEs or DAEs)

$$\phi\big(\dot{\boldsymbol{y}}(t), \boldsymbol{y}(t)\big) = \boldsymbol{0}, \quad \boldsymbol{y}, \phi \in \mathbb{R}^d. \tag{1.1}$$

It will be assumed that the initial conditions are consistent and that the IVP has a unique solution. Furthermore, we define the Jacobian matrices $K := \phi_{\boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{v})$ and $J := -\phi_{\boldsymbol{v}}(\boldsymbol{u}, \boldsymbol{v})$. In the case of *explicit* ordinary differential equations (ODEs) $\dot{\boldsymbol{y}}(t) = \boldsymbol{f}(\boldsymbol{y}(t))$ we have $\phi(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u} - \boldsymbol{f}(\boldsymbol{v})$, so that J denotes the Jacobian of the right-hand side function $\boldsymbol{f}$ of the ODE.

In this paper, we construct and analyze parallel iterative solvers for the solution of the sd-dimensional linear systems arising in the application of Newton's method to s-stage implicit Runge–Kutta (RK)

☆ The research reported in this paper was partially supported by the Technology Foundation (STW) in the Netherlands.

* Corresponding author.

PII S0168-9274(97)00071-8

type discretizations of (1.1). These linear solvers may be considered as *inner* iteration processes (the Newton process itself is the *outer* iteration process). The inner iteration process is partly an iterative method and partly a direct method. In fact, each inner iteration requires the solution of s linear subsystems, but now only of the IVP dimension d. We assume that direct solution methods are used for solving these subsystems. The computational effort consists mainly of the s LU-decompositions and secondly of the s forward–backward substitutions (briefly FBSs) needed in each inner iteration. As we will see, the LU-decompositions can be done in parallel, so that the *effective* costs on a parallel computer system with at least s processors are only one LU-decomposition of IVP dimension for each Jacobian update, yielding a considerable reduction of the wall-clock time (see Section 3 for reduction factors). As to the FBS-costs, we distinguish the Jacobi and the Gauss–Seidel approach. In the Jacobi approach, the s FBSs per inner iteration can be executed in parallel, so that the effective LU and FBS-costs only depend on the frequency of the Jacobian updates and the number of inner iterations, respectively, and not directly on the number of stages used in the RK discretization. For ODEIVPs, this Jacobi approach was used in [9,11]. We shall show that with appropriate changes, it can also be used in the IDE case. In the Gauss–Seidel approach (which is in fact a block Gauss–Seidel approach), part of the FBSs per inner iteration have to be done sequentially which increases the effective FBS costs.

The main purpose of this paper is a comparison of the convergence factors of the Jacobi and the Gauss–Seidel approach. The algorithmic properties of the inner iteration method are illustrated by solving an IDE of index 1 and of index 3 taken from the literature.

2. Runge–Kutta discretization

We define the Runge–Kutta type formula for solving the IDE (1.1) by (see, e.g., [4, p. 406])

$$\begin{aligned} &\boldsymbol{y}_{n+1} = (\boldsymbol{e}_s^{\mathrm{T}} \otimes I)\boldsymbol{Y}_{n+1}, \\ &\boldsymbol{R}_n(\boldsymbol{Y}_{n+1}) = 0, \qquad \boldsymbol{R}_n(\boldsymbol{Y}) := \Phi\big((h^{-1}A^{-1} \otimes I)(\boldsymbol{Y} - \boldsymbol{W}_n), \boldsymbol{Y}\big). \end{aligned} \tag{2.1}$$

Here, A denotes the s-by-s RK matrix which is assumed to be nonsingular, $\boldsymbol{W}_n$ is an sd-dimensional vector containing information from preceding steps, I is the d-by-d identity matrix, h is the stepsize $t_{n+1} - t_n$, and $\otimes$ denotes the Kronecker product. The s vector components $\boldsymbol{Y}_{n+1,i}$ of the sd-dimensional solution vector $\boldsymbol{Y}_{n+1}$ represent numerical approximations to the exact solution vectors $\boldsymbol{y}(t_n + c_i h)$, $\boldsymbol{c} = (c_i)$ being the abscissas vector with $c_s = 1$. Furthermore, $\boldsymbol{e}_s$ is the sth unit vector, $\boldsymbol{y}_n$ is the numerical approximation to $\boldsymbol{y}(t_n)$, and $\Phi(\boldsymbol{U}, \boldsymbol{V})$ contains the values $(\phi(\boldsymbol{U}_i, \boldsymbol{V}_i))$ for any pair of vectors $\boldsymbol{U} = (\boldsymbol{U}_i)$ and $\boldsymbol{V} = (\boldsymbol{V}_i)$. In the following, we denote with the symbol I the identity matrix, the dimension of which will be clear from the context.

The method (2.1) is completely specified by A, $\boldsymbol{W}_n$ and $\boldsymbol{c}$. If $\boldsymbol{W}_n = (E \otimes I)\boldsymbol{Y}_n$ with $E := \boldsymbol{e}\boldsymbol{e}_s^{\mathrm{T}}$, $\boldsymbol{e}$ representing the s-dimensional vector with unit entries, then (2.1) represents the one-step RK method $\{A, \boldsymbol{b}, \boldsymbol{c}\} = \{A, A^{\mathrm{T}}\boldsymbol{e}_s, \boldsymbol{c}\}$. Alternatives are the block methods where E is a matrix with eigenvalues on the unit disk, those of magnitude 1 being simple, and the k-step Radau methods where $\boldsymbol{W}_n$ is defined by a linear combination of the step point values $\boldsymbol{y}_n, \boldsymbol{y}_{n-1}, \ldots, \boldsymbol{y}_{n-k+1}$ (see, e.g., [4, p. 295]). In the following, most of our analysis applies to general back information vectors $\boldsymbol{W}_n$, but numerical illustrations will be confined to one-step Radau IIA methods.

The usual approach for solving the implicit equation $\boldsymbol{R}_n(\boldsymbol{Y}) = 0$ in (2.1) is the application of the modified Newton method

$$(I \otimes K_n - A \otimes hJ_n)\big(\boldsymbol{Y}^{(j)} - \boldsymbol{Y}^{(j-1)}\big) = -(hA \otimes I)\boldsymbol{R}_n\big(\boldsymbol{Y}^{(j-1)}\big), \quad j = 1, \ldots, m, \tag{2.2}$$

where K_n and J_n are the Jacobian matrices K and J evaluated at the step point t_n, and m is the number of Newton iterations which should be determined dynamically in an actual implementation. Each Newton iteration requires the solution of a linear system of dimension sd. However, already for moderate values of d, this is quite expensive, because the LU-decomposition of the sd-by-sd matrix $I \otimes K_n - A \otimes hJ_n$ requires as many as $\frac{2}{3}s^3d^3$ (real) arithmetic operations (operations counts will always refer to *real* arithmetic operations). This number of operations can be reduced by transforming (2.2) to "block-diagonal" form (cf. [3]). Let $\boldsymbol{Y}^{(j)} = (Q \otimes I)\widetilde{\boldsymbol{Y}}^{(j)}$, then (2.2) transforms to

$$\begin{aligned}&\big(I \otimes K_n - \widetilde{A} \otimes hJ_n\big)\big(\widetilde{\boldsymbol{Y}}^{(j)} - \widetilde{\boldsymbol{Y}}^{(j-1)}\big) = -\big(hQ^{-1}A \otimes I\big)\boldsymbol{R}_n\big((Q \otimes I)\widetilde{\boldsymbol{Y}}^{(j-1)}\big),\\ &\quad \widetilde{A} = Q^{-1}AQ, \quad j = 1, \ldots, m.\end{aligned} \tag{2.2'}$$

Assuming that A is nondefective, we can choose Q such that $\widetilde{A}$ is a σ-by-σ block-diagonal matrix with either one-by-one or two-by-two *real* diagonal blocks, where each diagonal block corresponds to an eigenvalue (pair) $\xi_k \pm i\eta_k$ of A. In fact,

$$\widetilde{A}_{kk} = \begin{cases} \xi_k, & \text{if } \eta_k = 0,\\ \begin{pmatrix} a_k & b_k \\ c_k & 2\xi_k - a_k \end{pmatrix}, & \text{if } \eta_k \neq 0. \end{cases} \tag{2.3a}$$

Here a_k, b_k and c_k are real parameters which depend on the matrix Q and which satisfy the relation

$$b_kc_k = -\big(a_k^2 - 2\xi_ka_k + \alpha_k^2\big), \quad \alpha_k := \sqrt{\xi_k^2 + \eta_k^2}. \tag{2.3b}$$

In the following it will always be assumed that $\xi_k > 0$ and that the ordering of the diagonal blocks $\widetilde{A}_{kk}$ is such that the ratio $|\eta_k/\xi_k|$ increases with k.

If the RK matrix A has only real eigenvalues, as is the case in the methods designed by Orel [12] and Bendtsen [1], then all diagonal blocks of $I \otimes K_n - \widetilde{A} \otimes hJ_n$ are of order d. When solving the linear system in (2.2′) by a direct linear solver, we need the LU-decompositions of these diagonal blocks, each requiring $\frac{2}{3}d^3$ operations. Hence, the *total* LU-costs are $\frac{2}{3}sd^3$ operations. However, since the LU-decompositions can be computed concurrently, the *effective* computational LU-costs in the block-diagonalized Newton method (2.2′) are only $\frac{2}{3}d^3$ operations, irrespective the value of s. Similarly, the FBSs can be performed in parallel.

A drawback of RK matrices with real eigenvalues is the relatively large value of s (and hence large numbers of processors) in order to achieve a given order of accuracy. More powerful methods with respect to order of accuracy and stability can be obtained by allowing A to have complex eigenvalues. For example, in the case of RK discretizations based on Gaussian quadrature formulas, A has at most one real eigenvalue (if s odd). Hence, the diagonal blocks of $I \otimes K_n - \widetilde{A} \otimes hJ_n$ are of order either d or $2d$, so that the LU-decompositions require either $\frac{2}{3}d^3$ or $\frac{16}{3}d^3$ operations. However, by writing the $2d$-dimensional systems as d-dimensional systems with complex coefficients, the LU-costs can be reduced to $\frac{8}{3}d^3$ operations (cf. Hairer and Wanner [4, p. 132]). Then, the *total* LU-costs are $\frac{2}{3}(2s-1)d^3$ operations for s odd and $\frac{2}{3}(2s)d^3$ operations for s even. Again, the LU-decompositions can be computed concurrently, so that the *effective* computational LU-costs are only $\frac{8}{3}d^3$ operations,

irrespective the value of s. The RADAUP codes of Hairer and Wanner [5] use the block-diagonalized Newton method applied to Radau IIA discretizations of (1.1).

Remark 2.1. In practice (see, e.g., [2]), it may be recommendable to remove the h^{-1} factor occurring in the residual function $\boldsymbol{R}_n(\boldsymbol{Y}^{(j-1)})$ by introducing "derivative" iterates $\dot{\boldsymbol{Y}}^{(j)}$ by the relation $\boldsymbol{Y}^{(j)} = \boldsymbol{W}_n + h(A \otimes I)\dot{\boldsymbol{Y}}^{(j)}$. Then, the iteration scheme (2.2) becomes

$$(I \otimes K_n - A \otimes hJ_n)\big(\dot{\boldsymbol{Y}}^{(j)} - \dot{\boldsymbol{Y}}^{(j-1)}\big) = -\boldsymbol{R}_n\big(\boldsymbol{W}_n + h(A \otimes I)\dot{\boldsymbol{Y}}^{(j-1)}\big), \quad j = 1, \ldots, m. \tag{2.4}$$

The sequences $\{\boldsymbol{Y}^{(j)}\}$ generated by the schemes (2.2) and (2.4) are algebraically identical, but (2.4) can be used as $h \to 0$. Furthermore, the structure of the Newton equations in (2.2) and (2.4) is similar.

3. Parallel linear system solvers

The usual approach for solving the linear systems in (2.2) is the application of the Butcher transformation to obtain the block-diagonalized Newton method (2.2′) and the application of a standard linear solver to the σ linear subsystems. The solution method analyzed in this paper is different and is characterized as follows:

(i) the matrix $\widetilde{A}$ in (2.2′) is allowed to be *block-triangular* with (real) diagonal blocks,
(ii) the linear subsystems of dimension d are solved by a standard linear solver,
(iii) the linear subsystems of dimension $2d$ are solved *iteratively* by a special *inner* iteration process based on splitting.

The diagonal blocks $\widetilde{A}_{kk}$ of $\widetilde{A}$ are again of the form (2.3) and such that the ratio $|\eta_k/\xi_k|$ increases with k. Furthermore, the inner iteration process is such that only linear systems of dimension are to be solved.

In the following, we shall consider both the case where $\widetilde{A}$ is block-diagonal and the case where $\widetilde{A}$ is block-triangular. The advantage of a block-triangular matrix $\widetilde{A}$ is that well-conditioned transformation matrices Q in the Butcher transformation $\boldsymbol{Y}^{(j)} = (Q \otimes I)\widetilde{\boldsymbol{Y}}^{(j)}$ can be chosen, so that there is no danger for amplification of iteration errors by an ill-conditioned back transformation. In Section 4, it will be shown that this improves the convergence factor associated with the inner iteration process.

3.1. Definition of the inner iteration process

Let $\widetilde{A} = \widetilde{C} + \widetilde{D}$, where $\widetilde{C}$ and $\widetilde{D}$ are the strictly lower block-triangular part and block-diagonal part of $\widetilde{A}$, respectively. Then, (2.2′) can be written as

$$\begin{aligned}
&\big(I \otimes K_n - \widetilde{D} \otimes hJ_n\big)\widetilde{\boldsymbol{Y}}^{(j)} = \boldsymbol{G}_n\big(\widetilde{\boldsymbol{Y}}^{(j)}, \widetilde{\boldsymbol{Y}}^{(j-1)}\big),\\
&\boldsymbol{G}_n\big(\widetilde{\boldsymbol{Y}}^{(j)}, \widetilde{\boldsymbol{Y}}^{(j-1)}\big) := \big(\widetilde{C} \otimes hJ_n\big)\widetilde{\boldsymbol{Y}}^{(j)} + \big(I \otimes K_n - \widetilde{A} \otimes hJ_n\big)\widetilde{\boldsymbol{Y}}^{(j-1)}\\
&\qquad\qquad - \big(hQ^{-1}A \otimes I\big)\boldsymbol{R}_n\big((Q \otimes I)\widetilde{\boldsymbol{Y}}^{(j-1)}\big).
\end{aligned} \tag{3.1}$$

This relation represents σ linear subsystems with system matrices $I \otimes K_n - \widetilde{A}_{kk} \otimes hJ_n$, $k = 1, \ldots, \sigma$, the dimension of which is either d or $2d$. Note that these systems have to be solved sequentially unless the matrix $\widetilde{C}$ vanishes. Let us partition $\widetilde{\boldsymbol{Y}}^{(j)}$ and $\boldsymbol{G}_n$ according to $\widetilde{\boldsymbol{Y}}^{(j)} = (\widetilde{\boldsymbol{y}}_1^{(j)}, \ldots, \widetilde{\boldsymbol{y}}_\sigma^{(j)})$ and $\boldsymbol{G}_n = (\boldsymbol{g}_{n1}, \ldots, \boldsymbol{g}_{n\sigma})$. Then the kth subsystem has the form

$$(I \otimes K_n - \widetilde{A}_{kk} \otimes hJ_n)\widetilde{\boldsymbol{y}}_k^{(j)} = \boldsymbol{g}_{nk}(\widetilde{\boldsymbol{Y}}^{(j-1)}, \widetilde{\boldsymbol{y}}_1^{(j)}, \ldots, \widetilde{\boldsymbol{y}}_{k-1}^{(j)}), \quad k = 1, \ldots, \sigma, \tag{3.1'}$$

where I has the same order as $\widetilde{A}_{kk}$. If $\widetilde{C} \neq O$, then the right-hand side is available as soon as the first $k-1$ subsystems have been solved. If $\widetilde{C} = O$, then the right-hand side does not depend on $(\widetilde{\boldsymbol{y}}_1^{(j)}, \ldots, \widetilde{\boldsymbol{y}}_{k-1}^{(j)})$, so that all subsystems can be solved concurrently.

The d-dimensional subsystems in the set (3.1′) are now solved by a standard linear solver, the $2d$-dimensional subsystems are solved iteratively by introducing the splitting $\widetilde{A}_{kk} = \widetilde{B}_{kk} + (\widetilde{A}_{kk} - \widetilde{B}_{kk})$, where $\widetilde{B}_{kk}$ is a diagonalizable 2-by-2 matrix with positive eigenvalues. This leads to the (inner) iteration method

$$\begin{aligned}&(I \otimes K_n - \widetilde{B}_{kk} \otimes hJ_n)\widetilde{\boldsymbol{y}}_k^{(j,\nu)} + ((\widetilde{B}_{kk} - \widetilde{A}_{kk}) \otimes hJ_n)\widetilde{\boldsymbol{y}}_k^{(j,\nu-1)} \\ &\quad = \boldsymbol{g}_{nk}(\widetilde{\boldsymbol{Y}}^{(j-1)}, \widetilde{\boldsymbol{y}}_1^{(j)}, \ldots, \widetilde{\boldsymbol{y}}_{k-1}^{(j)}),\end{aligned} \tag{3.2}$$

where $\nu = 1, 2, \ldots, r$ is the inner iteration index (the number of inner iterations r may depend on k). Each inner iteration again requires the solution of a linear system of dimension $2d$. However, since $\widetilde{B}_{kk}$ is assumed to be diagonalizable, the system matrix $I \otimes K_n - \widetilde{B}_{kk} \otimes hJ_n$ of this system can be block-diagonalized into two subsystems of dimension d. Thus, using the block-diagonalized version of (3.2), we only have to solve linear systems of the IVP dimension d.

The inner iteration subprocesses can be executed in parallel if $\widetilde{C} = O$ and should be done in sequence if $\widetilde{C} \neq O$. In fact, the $\widetilde{C} = O$ and $\widetilde{C} \neq O$ version of the iteration method (3.2) may respectively be considered as *Jacobi* and (block) *Gauss–Seidel type* methods. However, assuming that the d-dimensional subsystems are solved by a direct method, in both cases all LU-decompositions can be done in parallel, so that the effective LU-costs are $\frac{2}{3}d^3$ operations (irrespective the value of s), yielding a reduction by a factor 4 when compared with the $\frac{8}{3}d^3$ operations required by the solution of the d-dimensional complex subsystems in (2.2′). In the Gauss–Seidel version, the main sequential part (that is, the part that cannot be parallelized) consists of the sequential execution of the FBSs to solve the s subsystems of dimension d. If $\bar{r}$ is the *averaged* number of iterations needed to solve these subsystems, then $s\bar{r}$ linear system solves are required, i.e., $2s\bar{r}d^2$ operations. If there are only $2d$-dimensional subsystems in (3.2), then effectively $\frac{1}{2}s\bar{r}$ linear system solves, i.e., $s\bar{r}d^2$ operations, are required, whereas block-diagonalized Newton (with block-diagonal $\widetilde{A}$) effectively requires only $8d^2$ operations. Hence, if $\bar{r}$ is large, then the advantage of the reduced LU-costs is easily lost. Thus, the linear solver (3.2) is only effective if we can choose $\widetilde{B}_{kk}$ such that $\bar{r}$ is small. The choice of $\widetilde{B}_{kk}$ will be discussed in Section 4.1.

3.2. Back transformation

Relation (3.2) can be used for the convergence analysis of the inner iteration process in the transformed space, that is, the convergence of the iterates $\widetilde{\boldsymbol{y}}_k^{(j,\nu)}$. However, for a convergence analysis of the back transformed iterates it is more convenient to introduce the inner iterates

$$\boldsymbol{Y}^{(j,\nu)} = (Q \otimes I)\widetilde{\boldsymbol{Y}}^{(j,\nu)}, \quad \widetilde{\boldsymbol{Y}}^{(j,\nu)} := (\widetilde{\boldsymbol{y}}_1^{(j,\nu)}, \ldots, \widetilde{\boldsymbol{y}}_\sigma^{(j,\nu)}), \tag{3.3}$$

where $\widetilde{\boldsymbol{y}}_k^{(j,\nu)} = \widetilde{\boldsymbol{y}}_k^{(j)}$ if the kth subsystem in (3.1′) happens to have dimension d (i.e., if the corresponding eigenvalue of A is real). In terms of $\widetilde{\boldsymbol{Y}}^{(j,\nu)}$, the inner iteration process reads

$$\big(I \otimes K_n - \widetilde{B} \otimes hJ_n\big)\widetilde{\boldsymbol{Y}}^{(j,\nu)} + \big((\widetilde{B} - \widetilde{D}) \otimes hJ_n\big)\widetilde{\boldsymbol{Y}}^{(j,\nu-1)} = \boldsymbol{G}_n\big(\widetilde{\boldsymbol{Y}}^{(j)}, \widetilde{\boldsymbol{Y}}^{(j-1)}\big), \quad \nu = 1,2,\ldots, \tag{3.4}$$

where $\boldsymbol{G}_n$ is defined in (3.1) and $\widetilde{B}$ is the block-diagonal matrix with diagonal blocks $\widetilde{B}_{kk}$ with $\widetilde{B}_{kk} = \widetilde{A}_{kk} = \xi_k$ if $\widetilde{A}_{kk}$ is a one-by-one block. Defining the pair $\{B, C\} = \{Q\widetilde{B}Q^{-1}, Q\widetilde{C}Q^{-1}\}$, the back transformation of (3.4) reads

$$\begin{aligned}&(I \otimes K_n - B \otimes hJ_n)\big(\boldsymbol{Y}^{(j,\nu)} - \boldsymbol{Y}^{(j,\nu-1)}\big)\\ &\quad = (C \otimes hJ_n)\boldsymbol{Y}^{(j)} - \big(I \otimes K_n - (A - C) \otimes hJ_n\big)\boldsymbol{Y}^{(j,\nu-1)}\\ &\qquad + (I \otimes K_n - A \otimes hJ_n)\boldsymbol{Y}^{(j-1)} - (hA \otimes I)\boldsymbol{R}_n\big(\boldsymbol{Y}^{(j-1)}\big), \quad \nu = 1,2,\ldots.\end{aligned} \tag{3.5}$$

Estimates of the speed of convergence should be based on the iteration errors associated with (3.5), rather than on the iteration errors associated with (3.4).

The linear solver (3.5) will be called a *PILSRK method* (parallel iterative linear system method for RK discretizations) and the process {(2.2), (3.5)} will be called a *Newton–PILSRK method.*

3.3. Generalization

Having obtained the back transformation (3.5) of the inner iteration method (3.4), one may wonder whether this method can be generalized by using other matrices B and C than the pair $\{B, C\} = \{Q\widetilde{B}Q^{-1}, Q\widetilde{C}Q^{-1}\}$. Indeed, the PILSRK method (3.5) is a consistent iteration process for solving the linear Newton systems in (2.2) for *any pair* $\{B, C\}$ such that $I \otimes K_n - B \otimes hJ_n$ is invertible. However, in order to have a *practical* method, we should impose conditions on B and C. In the first place, we should of course require that B is similar to a diagonal matrix B^*, i.e., $B^* = S^{-1}BS$ is diagonal for some nonsingular matrix S. If this condition is satisfied, then we can diagonalize (3.5) by means of the Butcher transformation $\boldsymbol{Y}^{(j)} = (S \otimes I)\boldsymbol{X}^{(j)}$ to obtain

$$\begin{aligned}&\big(I \otimes K_n - B^* \otimes hJ_n\big)\big(\boldsymbol{X}^{(j,\nu)} - \boldsymbol{X}^{(j,\nu-1)}\big)\\ &\quad = \big(C^* \otimes hJ_n\big)\boldsymbol{X}^{(j)} - \big(I \otimes K_n - \big(A^* - C^*\big) \otimes hJ_n\big)\boldsymbol{X}^{(j,\nu-1)}\\ &\qquad + \big(I \otimes K_n - A^* \otimes hJ_n\big)\boldsymbol{X}^{(j-1)} - \big(hS^{-1}A \otimes I\big)\boldsymbol{R}\big(\boldsymbol{Y}^{(j-1)}\big), \quad \nu = 1,2,\ldots,\end{aligned} \tag{3.6}$$

where $A^* = S^{-1}AS$ and $C^* = S^{-1}CS$. First of all, we see that the diagonal structure allows us to decouple the LU-decomposition of the system matrix $I \otimes K_n - B^* \otimes hJ_n$ into s LU-decompositions of size d. Furthermore, if $C^* = O$, then each inner iteration step in (3.6) can be decoupled into s independent iteration steps which can be executed in parallel. If $C^* \neq O$, then (3.6) shows that we can decouple each inner iteration step in (3.6) into two or more independent iteration steps by imposing a special block structure on the matrices C^* and $A^* - C^*$. In this way, we can define a more general family of PILSRK$\{B, C\}$ methods which contains the method $\{B, C\} = \{Q\widetilde{B}Q^{-1}, Q\widetilde{C}Q^{-1}\}$ described in the preceding sections as a special case.

Remark 3.1. The Jacobi version of (3.5) can be considered as a conventional iteration method for linear systems based on the splitting

$$I \otimes K_n - A \otimes hJ_n = (I \otimes K_n - B \otimes hJ_n) + (B - A) \otimes hJ_n.$$

Remark 3.2. If only one inner iteration is performed and if we set $\boldsymbol{Y}^{(j,0)} = \boldsymbol{Y}^{(j-1)}$ and $\boldsymbol{Y}^{(j)} = \boldsymbol{Y}^{(j,1)}$, then the PILSRK method (3.5) reduces to

$$\big(I \otimes K_n - (B+C) \otimes hJ_n\big)\big(\boldsymbol{Y}^{(j)} - \boldsymbol{Y}^{(j-1)}\big) = -(hA \otimes I)\boldsymbol{R}_n\big(\boldsymbol{Y}^{(j-1)}\big). \tag{3.7}$$

This scheme may be considered as an "approximate" Newton scheme obtained by replacing in (2.2) the matrix A by $B+C$. If $B+C$ is similar to a lower triangular matrix with positive diagonal entries, then we can diagonalize (3.7), so that effectively only one FBS is required per outer iteration. The method (3.7) is related to the PDIRK and PTIRK methods proposed in [7,8] for ODEs. These PDIRK and PTIRK methods are obtained by replacing K_n with the identity matrix, by setting $C = O$, and by choosing for B either a diagonal matrix or a lower triangular matrix.

Remark 3.3. Even on *sequential* computers the diagonalized forms of the PILSRK methods (3.5) may be more efficient than the block-diagonalized Newton method. For example, if s is even, then the total LU-costs and FBS-costs associated with (3.5) respectively require $\frac{2}{3}sd^3$ and $2s\bar{r}d^2$ operations, whereas block-diagonalized Newton requires $\frac{4}{3}sd^3$ and $4sd^2$ operations. Hence, if $\bar{r} \leqslant 2$, then the PILSRK method does not require more FBS-costs, while its LU-costs are 2 times less expensive.

4. Convergence results

The convergence can be studied by analyzing the (exact) error recursion

$$\begin{aligned} &\boldsymbol{Y}^{(j,\nu)} - \boldsymbol{Y}^{(j)} = M\big(\boldsymbol{Y}^{(j,\nu-1)} - \boldsymbol{Y}^{(j)}\big), \\ &M := (I \otimes K_n - B \otimes hJ_n)^{-1}\big((A-B-C) \otimes hJ_n\big). \end{aligned} \tag{4.1}$$

Here, B and C may be any pair of matrices, but as already pointed out, the PILSRK method (3.5) is only a feasible method if the matrices B and C have an appropriate structure (see Section 3.3).

In the convergence analysis, we shall suppose that

(i) K_n is nonsingular,

(ii) $\{K_n, J_n\}$ has a complete (generalized) eigensystem.

(We will refer to these assumptions as property P.) In practice, this is of course an unrealistic situation. However, by observing that d-by-d matrix pairs $\{K, J\}$ having property P are dense in the space of all d-by-d matrix pairs, we can define a one-parameter family of matrix pairs $\{K(\varepsilon), J(\varepsilon)\}$ which satisfies property P for $\varepsilon > 0$ and which converges to $\{K_n, J_n\}$ as $\varepsilon \to 0$. Hence, for the matrix $M(\varepsilon)$ corresponding to $\{K(\varepsilon), J(\varepsilon)\}$ we have $M(\varepsilon) \to M$ as $\varepsilon \to 0$. Thus, a convergence analysis based on property P is relevant for the case where this property is not satisfied.

A necessary and sufficient condition for convergence of the PILSRK methods is $\rho(M) < 1$. In order to obtain the eigenvalues of M, we shall list all its eigenvectors. First we look for eigenvectors of the form $\boldsymbol{a} \otimes \boldsymbol{w}$, where $\boldsymbol{a}$ and $\boldsymbol{w}$ are vectors of dimensions s and d, respectively. If the eigenvalues are denoted by μ, then $h(A-B-C+\mu B)\boldsymbol{a} \otimes J_n\boldsymbol{w} = \mu \boldsymbol{a} \otimes K_n\boldsymbol{w}$. This shows that $J_n\boldsymbol{w}$ and $K_n\boldsymbol{w}$ are related by the eigenvalue equation $J_n\boldsymbol{w} = \lambda K_n\boldsymbol{w}$, i.e., λ is a (generalized) eigenvalue of the matrix pair $\{K_n, J_n\}$. On substitution of $J_n\boldsymbol{w} = \lambda K_n\boldsymbol{w}$ and by defining $z := \lambda h$, we obtain

$$z(A-B-C)\boldsymbol{a} \otimes K_n\boldsymbol{w} = \mu(I - zB)\boldsymbol{a} \otimes K_n\boldsymbol{w}.$$

Since $K_n \boldsymbol{w} \neq 0$, $\mu = \mu(z)$ is an eigenvalue of the amplification matrix

$$Z(z) := z(I - zB)^{-1}(A - B - C). \tag{4.2}$$

We now impose the condition that $Z(h\lambda)$ is nondefective for all λ in the spectrum of $\sigma(K_n, J_n)$. This condition and condition (ii) of property P imply that M has sd eigenvectors of the form $\boldsymbol{a} \otimes \boldsymbol{w}$. Hence, *all* eigenvectors of M are of this form and its eigenvalues are given by those of $Z(h\lambda)$ with $\lambda \in \sigma(K_n, J_n)$.

Let $\rho(z)$ be the spectral radius of $Z(z)$. Then, the *region of convergence* is defined by the region

$$\Gamma := \{z\colon\ \rho(z) < 1,\ Z(z) \text{ is nondefective}\}.$$

In analogy with the terminology used in the linear stability theory, we shall call the PILSRK method *A-convergent* if Γ contains the left halfplane and *L-convergent* if it is A-convergent and if $\rho(z)$ vanishes at infinity. Matrix pairs $\{B, C\}$ will be said to lie in the set $\mathbb{B}(A)$ if $\{B, C\}$ generates an A-convergent PILSRK method.

From the considerations above we conclude that the PILSRK method (3.5) converges for all $h > 0$ if $\{B, C\} \in \mathbb{B}(A)$, if $\{K_n, J_n\}$ satisfies property P and if its spectrum is in the nonpositive halfplane.

In order to get some insight into the convergence behaviour, we observe that after ν iterations the eigenvector components of the iteration error are amplified by $Z^\nu(z)$, where $z = h\lambda$ corresponds with the (generalized) eigenvectors of $\{K_n, J_n\}$. Let us define the *averaged amplification factor* by

$$\rho^{(\nu)}(z) := \sqrt[\nu]{\|Z^\nu(z)\|}. \tag{4.3}$$

Evidently, the spectral radius $\rho(z)$ of $Z(z)$ equals $\rho^{(\infty)}(z)$ and will therefore be called the *asymptotic* amplification factor.

4.1. The asymptotic amplification factor

We shall now consider the case $\{B, C\} = \{Q\widetilde{B}Q^{-1}, Q\widetilde{C}Q^{-1}\}$, i.e., the process (3.4), in more detail. We first derive estimates for the asymptotic amplification factor $\rho(z) = \rho^{(\infty)}(z)$. Since the matrix $\widetilde{Z}(z) = QZ(z)Q^{-1}$ is block-diagonal with either one-by-one or two-by-two diagonal blocks, we may confine our considerations to the diagonal blocks of $\widetilde{Z}(z)$. Let us define the one-by-one blocks $\widetilde{B}_{kk}$ of $\widetilde{B}$ by ξ_k (as in (3.4)) and the two-by-two diagonal blocks

$$\widetilde{B}_{kk} := \begin{pmatrix} u_k & x_k \\ v_k & w_k \end{pmatrix}, \qquad u_k + w_k > 2\sqrt{u_k w_k - x_k v_k}, \quad u_k w_k - x_k v_k > 0 \tag{4.4}$$

(the conditions on the entries of $\widetilde{B}_{kk}$ ensure that its eigenvalues are distinct and positive, so that $\widetilde{B}$ is diagonalizable). We shall require that $\{B, C\}$ generates an L-convergent iteration method. This requirement is crucial in order to quickly remove stiff components from the iteration error (see [7]). In fact, L-convergence implies that $Z^\nu(\infty)$ vanishes for $\nu \geqslant 2$, because the matrix $\widetilde{Z}(z)$ is block-diagonal. Hence, within two inner iterations, all stiff error components are removed from the iteration error.

The following result provides a lower bound for $\rho(z)$ in the left halfplane when we impose the condition of L-convergence. The proof parallels the proof of a similar theorem in [9].

Theorem 4.1. *Let* $\{B,C\} = \{Q\widetilde{B}Q^{-1}, Q\widetilde{C}Q^{-1}\}$, *let* $\widetilde{A}_{kk}$ *and* $\widetilde{B}_{kk}$ *be defined by* (2.3) *and* (4.4), *and let the generated PILSRK method be L-convergent. Then, we have in the left halfplane for all* a_k *and* b_k *the inequality*

$$\rho := \max_{\mathrm{Re}(z)\leqslant 0} \rho(z) \geqslant 1 - \frac{\xi_\sigma}{\alpha_\sigma}.$$

Proof. The eigenvalues $\mu_k(z)$ of the matrix $Z(z) = Q\widetilde{Z}(z)Q^{-1}$ are given by those of the diagonal blocks

$$\widetilde{Z}_{kk}(z) := z\big(I - z\widetilde{B}_{kk}\big)^{-1}\big(\widetilde{A}_{kk} - \widetilde{B}_{kk}\big). \tag{4.5}$$

If $\eta_k = 0$, then $\widetilde{Z}_{kk}(z)$ vanishes yielding zero eigenvalues. Therefore, we may restrict our considerations to the two-by-two diagonal blocks of $\widetilde{Z}(z)$. The eigenvalues of these blocks satisfy the characteristic equation

$$\det\begin{pmatrix} a_k - u_k - \big(z^{-1} - u_k\big)\mu_k(z) & b_k - x_k + x_k\mu_k(z) \\ c_k - v_k + v_k\mu_k(z) & 2\xi_k - a_k - w_k - \big(z^{-1} - w_k\big)\mu_k(z) \end{pmatrix} = 0, \quad \eta_k \neq 0. \tag{4.6}$$

It is easily verified that $\mu_k(z)$ vanishes at infinity if

$$v_k = \frac{(a_k - 2\xi_k)u_k^2 + (2\alpha_k^2 + c_k x_k)u_k - a_k\alpha_k^2}{a_k x_k - b_k u_k}, \qquad w_k = \frac{\alpha_k^2 + x_k v_k}{u_k}, \tag{4.7}$$

where x_k and u_k are still free. In addition, we have to satisfy the inequalities in (4.4). Since $u_k w_k = \alpha_k^2 + x_k c_k$, these inequalities are satisfied if $u_k + w_k > 2\alpha_k$. Eq. (4.6) is solved by

$$\mu_k(z) = 0, \qquad \mu_k(z) = \frac{(2\xi_k - u_k - w_k)z}{\alpha_k^2 z^2 - (u_k + w_k)z + 1}. \tag{4.8}$$

Since the function $\mu_k(z)$ is regular in the left halfplane, its maximum in the left halfplane is assumed on the imaginary axis. It is easily verified that

$$\big|\mu_k(\mathrm{i}y)\big| = \frac{|2\xi_k - u_k - w_k|\,|y|}{\sqrt{(1 - \alpha_k^2 y^2)^2 + (u_k + w_k)^2 y^2}} \tag{4.9}$$

assumes an absolute maximum at $y = \pm\alpha_k^{-1}$ which is given by

$$\max_{\mathrm{Re}(z)\leqslant 0} \rho\big(\widetilde{Z}_{kk}(z)\big) = \left|1 - \frac{2\xi_k}{u_k + w_k}\right|, \quad u_k + w_k > 2\alpha_k. \tag{4.10}$$

This value is bounded below by $1 - \xi_k\alpha_k^{-1}$ (we recall that we have assumed $\xi_k > 0$). Hence, we see that the eigenvalues of A with the smallest ratio $\xi_k\alpha_k^{-1}$, that is, the eigenvalues with the relatively largest imaginary part, determine a lower bound for $\rho(z)$. Recalling that the ordering of the diagonal blocks $\widetilde{A}_{kk}$ is such that $|\eta_k/\xi_k|$ increases with k, we conclude that in the left halfplane the maximal value of $\rho(z)$ is bounded below by $|1 - \xi_\sigma\alpha_\sigma^{-1}|$, proving the assertion of the theorem. □

From (4.10) it is clear that the best we can do is to choose $u_k + w_k = 2\theta_k\alpha_k$, where $\theta_k = 1 + \varepsilon_k$ with ε_k a small positive number. By virtue of (4.7) we obtain the relation

$$u_k^2 + \alpha_k^2 + x_k \frac{(a_k - 2\xi_k)u_k^2 + (2\alpha_k^2 + c_k x_k)u_k - a_k\alpha_k^2}{a_k x_k - b_k u_k} = 2\theta_k\alpha_k u_k.$$

This equation shows that by choosing $x_k = 0$, we can compute u_k *independently* of the values a_k, b_k and c_k. Since the block-triangularizing matrix Q depends on a_k, b_k, c_k, and vice versa, we preserve a maximal amount of freedom in selecting Q if $x_k = 0$. Setting $x_k = 0$ and introducing the new parameter

$$\gamma_k := \theta_k \pm \sqrt{\theta_k^2 - 1} \quad \text{with } \theta_k > 1,$$

we obtain

$$u_k = \gamma_k\alpha_k, \qquad v_k = c_k\alpha_k \frac{(\gamma_k^2 - 1)a_k - 2\gamma_k^2\xi_k + 2\gamma_k\alpha_k}{\gamma_k(a_k^2 - 2\xi_k a_k + \alpha_k^2)}, \qquad w_k = \frac{\alpha_k}{\gamma_k}. \tag{4.11}$$

We remark that this expression holds for both signs in the formula for γ_k. By virtue of (4.10), we have the result

$$\max_{\mathrm{Re}(z)\leqslant 0} \rho\big(\widetilde{Z}_{kk}(z)\big) = 1 - \frac{2\gamma_k\xi_k}{(\gamma_k^2 + 1)\alpha_k}, \qquad \gamma_k \neq 1, \quad \gamma_k > 0. \tag{4.10′}$$

Furthermore, it follows from (4.8) that firstly, $Z(z)$ has a complete eigensystem for all finite z (this is also true for $z = 0$, because $Z(0) = O$), and secondly, $\rho(\widetilde{Z}_{kk}(z)) < 1$ in the region

$$\big(2\gamma_k\xi_k - \gamma_k^2\alpha_k - \alpha_k\big)^2|z|^2 < \big|\gamma_k\alpha_k^2 z^2 - \big(\gamma_k^2\alpha_k + \alpha_k\big)z + \gamma_k\big|^2. \tag{4.12}$$

Thus, we have proved

Theorem 4.2. *For $\gamma_k \neq 1$, $\gamma_k > 0$, $x_k = 0$ and all a_k, b_k and c_k satisfying* (2.3b) *we have*

(i) *the matrix pair $\{B, C\} = \{Q\widetilde{B}Q^{-1}, Q\widetilde{C}Q^{-1}\}$ is in $\mathbb{B}(A)$ if $\widetilde{B}$ satisfies* (4.11),

(ii) *the convergence region Γ is given by the cross section of the regions* (4.12),

(iii) $$\max_{\mathrm{Re}(z)\leqslant 0} \rho(z) = \max_k \rho_k, \quad \rho_k := 1 - \frac{2\gamma_k\xi_k}{(\gamma_k^2 + 1)\alpha_k}.$$

A comparison with Theorem 4.1 shows that values of γ_k close to 1 yield an "almost" minimal spectral radius. In Table 1, we have listed for a few Radau IIA methods the values of ρ_k in the case $\gamma_k = 7/8$. These values are worst-case values, because in the greater part of the left halfplane, $\rho(z)$ is much smaller.

Table 1
Values of ρ_k for s-stage Radau IIA methods ($\gamma_k = 7/8$)

k	$s = 2$	$s = 4$	$s = 6$	$s = 8$
1	0.19	0.06	0.03	0.02
2		0.45	0.21	0.12
3			0.57	0.33
4				0.64

4.2. The averaged amplification factors

First we compute the averaged amplification factor for the eigenvector components of the iteration error associated with (3.4) after ν iterations. These components are amplified by $\widetilde{Z}_{kk}^{\nu}(z)$, where $z = h\lambda$ corresponds with the (generalized) eigenvectors of $\{K_n, J_n\}$. Hence, the averaged amplification factor associated with (3.4) is defined by

$$\tilde{\rho}^{(\nu)}(z) := \max_k \tilde{\rho}_k^{(\nu)}(z), \quad \tilde{\rho}_k^{(\nu)}(z) := \sqrt[\nu]{\left\|\widetilde{Z}_{kk}^{\nu}(z)\right\|}.$$

Here, $\tilde{\rho}_k^{(\nu)}(z)$ may be considered as the averaged amplification factor associated with (3.2). Let us set the free parameter $\gamma_k = a_k\alpha_k^{-1}$, so that the matrix $\widetilde{Z}_{kk}^{\nu}(z)$ simplifies to (cf. [9])

$$\widetilde{Z}_{kk}^{\nu}(z) = \mu_k^{\nu}(z)\begin{pmatrix} 0 & q_k(z) \\ 0 & 1 \end{pmatrix},$$

$$\mu_k(z) := \frac{b_k c_k z}{(1 - a_k z)(a_k - \alpha_k^2 z)}, \quad q_k(z) := \frac{a_k - \alpha_k^2 z}{c_k}. \tag{4.13}$$

With respect to the Euclidean norm, we have

$$\tilde{\rho}_k^{(\nu)}(z) = \left|\mu_k(z)\right|\left(1 + \left|q_k(z)\right|^2\right)^{1/2\nu} = \left|\mu_k(z)\right|^{1-1/\nu}\left(\left|\mu_k(z)\right|^2 + \left|\mu_k(z)q_k(z)\right|^2\right)^{1/2\nu}.$$

We majorize this expression in the left halfplane by using the maximum values of $|\mu_k(z)|$ and the maximum value of $|\mu_k(z)q_k(z)|$. From Theorem 4.2 it follows that $|\mu_k(z)| \leqslant \rho_k$ and an elementary calculation yields

$$\left|\mu_k(z)\right|^2 + \left|\mu_k(z)q_k(z)\right|^2 \leqslant \rho_k^2 + b_k^2 a_k^{-2}.$$

Thus,

$$\tilde{\rho}_k^{(\nu)}(z) \leqslant \rho_k \sqrt[2\nu]{\beta_k}, \quad \beta_k = 1 + \frac{b_k^2}{a_k^2 \rho_k^2}. \tag{4.14}$$

Expressing the upper bound (4.14) in terms of the parameters a_k, b_k and c_k, we obtain for the amplification factors in the transformed and untransformed space the estimates given in the theorem:

Theorem 4.3. *If $\gamma_k = a_k\alpha_k^{-1}$, then with respect to the Euclidean norm, the averaged amplification factors $\tilde{\rho}_k^{(\nu)}(z)$ and $\rho^{(\nu)}(z)$ satisfy in the left halfplane the inequalities*

$$\begin{aligned} &\tilde{\rho}_k^{(\nu)}(z) \leqslant \rho_k \sqrt[2\nu]{\beta_k}, \qquad \rho_k = 1 - \frac{2a_k\xi_k}{a_k^2 + \alpha_k^2}, \quad \beta_k = 1 + \frac{(a_k^2 + \alpha_k^2)^2}{a_k^2 c_k^2}, \\ &\rho^{(\nu)}(z) := \sqrt[\nu]{\left\|Z^{\nu}(z)\right\|} \leqslant \sqrt[\nu]{\kappa(Q)}\max_k \tilde{\rho}_k^{(\nu)}(z), \quad \kappa(Q) := \|Q\|\left\|Q^{-1}\right\|. \end{aligned} \tag{4.15}$$

4.3. The transformation matrix

Theorem 4.3 suggests the use of transformation matrices Q such that $a_k \approx \alpha_k$ (to achieve that ρ_k is close to its minimal value) and $c_k^2 \gg 1$. Such transformation matrices can be constructed, however, they turn out to be poorly conditioned (cf. [9]), so that we have fast convergence in the transformed

space, but not necessarily fast convergence in the untransformed space. Therefore, we shall restrict our considerations to *orthogonal* transformation matrices. This excludes the Jacobi case with $C = O$, so that we should consider the Gauss–Seidel case $C \neq O$ (the Jacobi case $\{B, C\} = \{Q\tilde{B}Q^{-1}, Q\}$ with non-orthogonal Q has been analyzed in [9]).

In order to construct an orthogonal matrix Q, let R be an orthogonal transformation matrix such that $\underline{S} := R^{-1}AR$ is a real Schur form of A with two-by-two diagonal blocks given by

$$\underline{S}_{kk} = \xi_k, \qquad \text{if } \eta_k = 0,$$
$$\underline{S}_{kk} = \begin{pmatrix} \underline{a}_k & \underline{b}_k \\ \underline{c}_k & 2\xi_k - \underline{a}_k \end{pmatrix}, \quad \underline{b}_k\underline{c}_k = -\left(\underline{a}_k^2 - 2\xi_k\underline{a}_k + \alpha_k^2\right), \quad \text{if } \eta_k \neq 0.$$

The values of $\underline{a}_k$ and $\underline{b}_k$ are completely determined by R (for the construction of R we refer to [11]). We now transform these diagonal blocks by a block-diagonal rotation matrix $\Delta = (\Delta_{jk})$ with

$$\Delta_{kk} = \begin{cases} 1, & \text{if } \eta_k = 0, \\ \begin{pmatrix} \cos(\psi_k) & -\sin(\psi_k) \\ \sin(\psi_k) & \cos(\psi_k) \end{pmatrix}, & \text{if } \eta_k \neq 0. \end{cases}$$

Then, $Q = R\Delta$, $\kappa(Q) = 1$ and $\tilde{A} = \Delta^{-1}R^{-1}AR\Delta$, where the diagonal blocks of $\tilde{A}$ are given by (2.3) with

$$a_k = \underline{a}_k\cos^2(\psi_k) - (\underline{b}_k + \underline{c}_k)\cos(\psi_k)\sin(\psi_k) + (2\xi_k - \underline{a}_k)\sin^2(\psi_k),$$
$$c_k = \underline{c}_k\cos^2(\psi_k) + 2(\underline{a}_k - \xi_k)\cos(\psi_k)\sin(\psi_k) - \underline{b}_k\sin^2(\psi_k).$$

The parameter ψ_k is chosen such that the spectral radius ρ_k occurring in the upper bound (4.15) is minimized. This means that a_k should be close to α_k. In order to avoid defective matrices $\tilde{B}_{kk}$, we should not allow $a_k = \alpha_k$. Let us impose the constraints $a_k \leqslant 7\alpha_k/8$ and $a_k \geqslant 9\alpha_k/8$. We now determine ψ_k such that $|a_k - \alpha_k|$ is minimized subject to these constraints. If more than one values of ψ_k are found that minimizes $|a_k - \alpha_k|$, then we take the value that also minimizes β_k.

For the s-stage Radau IIA methods with $s = 2$, 4, 6 and 8, we found that there exists ψ_k-values such that $a_k = 7\alpha_k/8$ (i.e., $\gamma_k = 7/8$). The corresponding ρ_k-values are listed in Table 1. For $s = 4$ and $s = 8$, the corresponding β_k-values are respectively given by $\{3.3, 2.0\}$ and $\{2.6, 2.8, 2.4, 2.4\}$.

Table 2 lists the *actual* left halfplane upper bounds for $\rho^{(\nu)}(z)$ using the Euclidean norm in its definition (4.3) (in brackets, we listed for $\tilde{C} \neq O$ the theoretical upper bounds of (4.15), which are

Table 2
Actual upper bounds for $\rho^{(\nu)}(z)$ for Radau IIA ($\gamma_k = 7/8$)

	$s = 4$		$s = 8$	
ν	Jacobi	Gauss–Seidel	Jacobi	Gauss–Seidel
1	1.95	0.54 (0.63)	3.51	0.84 (0.98)
2	0.98	0.49 (0.53)	1.88	0.73 (0.80)
3	0.76	0.48 (0.50)	1.30	0.70 (0.74)
4	0.66	0.47 (0.49)	1.19	0.68 (0.71)
⋮	⋮	⋮	⋮	⋮
∞	0.45	0.45 (0.45)	0.64	0.64 (0.64)

too pessimistic by only less than 15%). The amplification factors for $\widetilde{C} = O$ are taken from [9] and turn out to be considerably larger.

4.4. *Effect of the predictor*

The averaged amplification factor $\rho^{(\nu)}(z)$ defined in (4.3) does not take the amplification effect of the predictor formula for $\boldsymbol{Y}^{(0)}$ needed in (3.5) into account. This effect plays a role in the overall convergence. For example, the predictor may have or may have not a strong damping effect on the stiff iteration error components. It can be included by modifying the definition of the convergence factor. To that end, we first have to specify the predictors we are going to use. Let $\boldsymbol{Y}^{(1,0)} = \boldsymbol{Y}^{(0)}$ with $\boldsymbol{Y}^{(0)}$ defined by either

$$\Phi\big((h^{-1}[B+C]^{-1} \otimes I)(\boldsymbol{Y}^{(0)} - (E \otimes I)\boldsymbol{Y}_n), \boldsymbol{Y}^{(0)}\big) = 0 \tag{4.16a}$$

or

$$\boldsymbol{Y}^{(0)} = (E_P \otimes I)\boldsymbol{Y}_n, \tag{4.16b}$$

where $\boldsymbol{Y}_n$ is the stage vector computed in the preceding step and the matrix E_P can be used to control the order of accuracy of $\boldsymbol{Y}^{(0)}$. The second predictor formula (4.16b) is an extrapolation formula based on the back values contained in the preceding stage vector $\boldsymbol{Y}_n$. The first predictor formula (4.16a) is obtained from (2.1) by replacing A with $B + C$ and $\boldsymbol{W}_n$ with $(E \otimes I)\boldsymbol{Y}_n$. This formula can be solved by a modified Newton process (2.2) using the predictor formula (4.16b) to start the iteration process. Performing only one outer iteration, this predictor formula becomes

$$\begin{aligned}
&(I \otimes K_n - (B+C) \otimes hJ_n)(\boldsymbol{Y}^{(j)} - \boldsymbol{Y}^{(j-1)}) \\
&\quad = -h((B+C) \otimes I)\Phi\big((h^{-1}[B+C]^{-1} \otimes I)(\boldsymbol{Y}^{(j-1)} - (E \otimes I)\boldsymbol{Y}_n), \boldsymbol{Y}^{(j-1)}\big), \\
&j = 1, \ldots, m,
\end{aligned}$$

where $\boldsymbol{Y}^{(0)} = (E_P \otimes I)\boldsymbol{Y}_n$. Evidently, the same LU decompositions as in (3.5) can be used, so that only FBSs are needed.

We consider the predictor effect for *linear* problems and for $E := \boldsymbol{e}\boldsymbol{e}_s^{\mathrm{T}}$. Let P be a matrix which equals either $B + C$ or O. Then, we may write the residual function (2.1) and the predictors (4.16) as

$$\boldsymbol{R}_n(\boldsymbol{Y}) = (h^{-1}A^{-1} \otimes I)\big((I \otimes K_n - A \otimes hJ_n)\boldsymbol{Y} - (E \otimes K_n)\boldsymbol{Y}_n\big),$$

$$(I \otimes K_n - P \otimes hJ_n)\boldsymbol{Y}^{(0)} = (E_P \otimes K_n)\boldsymbol{Y}_n,$$

respectively. From (4.1) and (2.2) it follows that

$$\begin{aligned}
\boldsymbol{Y}^{(1,\nu)} - \boldsymbol{Y}^{(1)} &= M^{\nu}\big(\boldsymbol{Y}^{(1,0)} - \boldsymbol{Y}^{(1)}\big) = M^{\nu}\big(\boldsymbol{Y}^{(0)} - \boldsymbol{Y}^{(1)}\big) \\
&= M^{\nu}(I \otimes K_n - A \otimes hJ_n)^{-1}(hA \otimes I)\boldsymbol{R}_n\big(\boldsymbol{Y}^{(0)}\big) \\
&= M^{\nu}\big(\boldsymbol{Y}^{(0)} - (I \otimes K_n - A \otimes hJ_n)^{-1}(E \otimes K_n)\boldsymbol{Y}_n\big) \\
&= M^{\nu}\big((I \otimes K_n - P \otimes hJ_n)^{-1}(E_P \otimes K_n) \\
&\quad - (I \otimes K_n - A \otimes hJ_n)^{-1}(E \otimes K_n)\big)\boldsymbol{Y}_n.
\end{aligned} \tag{4.17}$$

Table 3
Actual upper bounds for $\rho_{\text{pred}}^{(\nu)}(z)$ for 4-stage Radau IIA with $\gamma_k = 7/8$

	Jacobi			Gauss–Seidel		
ν	LSV ($q=0$)	EPL ($q=3$)	GLM ($q=2$)	LSV ($q=0$)	EPL ($q=3$)	GLM ($q=2$)
1	2.66	6.98	3.48	0.92	39.5	2.93
2	1.15	2.27	1.30	0.61	4.17	0.88
3	0.84	1.34	0.91	0.54	1.97	0.67
4	0.71	1.02	0.76	0.51	1.36	0.61
⋮	⋮	⋮	⋮	⋮	⋮	⋮
∞	0.45	0.45	0.45	0.45	0.45	0.45

Table 4
Total and effective LU- and FBS-costs for even or odd numbers of RK stages

Method	Total LU/d^3	Total FBS/d^2	Eff. LU/d^3	Eff. FBS/d^2
Block-diagonalized Newton	$\frac{4s}{3}$ or $\frac{2}{3}(2s-1)$	$4s$ or $4s-2$	$\frac{8}{3}$	8
PILSRK: Gauss–Seidel	$\frac{2s}{3}$	$2s\bar{r}$ or $2(s\bar{r}_2-\bar{r}_2+1)$	$\frac{2}{3}$	$s\bar{r}$ or $s\bar{r}_2-\bar{r}_2+2$
PILSRK: Jacobi	$\frac{2s}{3}$	$2sr$	$\frac{2}{3}$	$2r$

Taking into account the computational effort involved in applying the predictor formula, we are led to the following definition of the averaged amplification factor associated with (4.17):

$$\rho_{\text{pred}}^{(\nu)}(z) := \sqrt[\nu]{\left\| Z^{\nu-\theta}(z)Z^*(z) \right\|}, \quad Z^*(z) := (I-zP)^{-1}E_P - (I-zA)^{-1}E, \tag{4.18}$$

where $\theta = 1$ if $P = B + C$ and $\theta = 0$ if $P = O$.

A qth-order accurate predictor is obtained by defining E_P according to

$$E_P\boldsymbol{e} = \boldsymbol{e}, \qquad E_P X = U - PV,$$

$$U := \left(\boldsymbol{c}^j/j\right), \quad V := \left(\boldsymbol{c}^{j-1}\right), \quad X := \left((\boldsymbol{c}-\boldsymbol{e})^j/j\right), \qquad j = 1,\ldots,q. \tag{4.19}$$

There are various options for choosing E_P. For $P = O$, we have considered the case $E_P = \boldsymbol{e}\boldsymbol{e}_s^{\mathrm{T}}$ (last step value (LSV) predictor) and the case where E_P is defined according to (4.19) with $q = s-1$ (maximal order extrapolation (EPL)). For $P = B + C$, we defined E_P according to (4.19) with $q = 2$ and we used the remaining free parameters to minimize both $\rho_{\text{pred}}^{(1)}(z) + \rho_{\text{pred}}^{(2)}(z)$ in the left halfplane (GLM predictor).

Table 3 lists left halfplane upper bounds for $\rho_{\text{pred}}^{(\nu)}(z)$ in the case of the 4-stage Radau IIA method. In appreciating these values, we should take the effect of the *order of accuracy* of the predictor into account. For example, the LSV predictor together with the Gauss–Seidel version $\widetilde{C} \neq O$ possesses the smallest left halfplane upper bounds for $\rho_{\text{pred}}^{(\nu)}(z)$, but its zero order will be a drawback (see Section 5).

4.5. Comparison of LU- and FBS-costs

We conclude this section by summarizing the total and effective LU-costs per Jacobian update and the total and effective FBS-costs per outer iteration. Table 4 lists these costs for the block-diagonalized

Newton, the Newton–PILSRK method derived above, and Newton–PILSRK methods with $C = O$. The cost formulas are given for the cases s even and s odd, assuming that the matrix A has no real eigenvalues if s is even and only one real eigenvalue if s is odd. In the case of the PILSRK $\{Q\widetilde{B}Q^{-1}, Q\widetilde{C}Q^{-1}\}$ method, $\overline{r}$ and $\overline{r}_2$ respectively denote the averaged number of inner iterations over *all* subsystems and over subsystems associated with the *complex* eigenvalue pairs of A. Finally, r denotes the maximal number of inner iterations needed for the s subsystems in the PILSRK $\{B, O\}$ method. The figures in Table 4 show that the two PILSRK methods require the same number of (total and effective) LU operations. Their effective FBS-costs are highly dependent on the value of $\overline{r}$, $\overline{r}_2$ and r.

5. Numerical experiments

The aim of this section is to compare the algorithmic properties of the Newton–PILSRK method $\{(2.2), (3.5)\}$. We compare the Gauss–Seidel version $\widetilde{C} \neq O$ with Q orthogonal as analyzed in this paper and the Jacobi version $\widetilde{C} = O$ with Q nonorthogonal analyzed in [9]. In both cases, we take $\gamma_k = 7/8$. The comparisons are carried out for a few test problems from the literature.

The corrector, i.e., the matrix A, is defined by the 4-stage Radau IIA corrector and the predictor formula is either the LSV or the EPL predictor (see Section 4.4), and is specified in the tables of results.

In the Jacobi case, we have (cf. [9])

$$B = \begin{pmatrix} 0.1096 & -0.0430 & 0.0268 & -0.0080 \\ 0.2085 & 0.3064 & -0.0671 & 0.0211 \\ 0.2484 & 0.0823 & 0.2573 & -0.0142 \\ 0.2596 & -0.0515 & 0.4219 & 0.0780 \end{pmatrix}, \qquad C = O, \tag{5.1a}$$

and in the Gauss–Seidel case

$$B = \begin{pmatrix} 0.1175 & -0.0207 & 0.0255 & -0.0017 \\ 0.2555 & 0.2758 & -0.0535 & 0.0037 \\ -0.0256 & -0.0076 & 0.2030 & -0.0002 \\ 0.0206 & 0.0528 & 0.3488 & 0.1549 \end{pmatrix},$$

$$C = \begin{pmatrix} -0.0061 & -0.0117 & 0.0002 & -0.0019 \\ -0.0281 & -0.0400 & -0.0002 & -0.0059 \\ 0.2492 & 0.3955 & -0.0010 & 0.0614 \\ 0.2099 & 0.3114 & 0.0007 & 0.0470 \end{pmatrix}. \tag{5.2a}$$

Diagonalizing (3.5) by the transformation $\mathbf{Y}^{(j)} = (S \otimes I)\mathbf{X}^{(j)}$ yields the method (3.6) with

$$S = \begin{pmatrix} 2.9526 & 0.3159 & 1.5325 & 0.0276 \\ -7.2663 & -0.8756 & -1.0553 & -0.3113 \\ 3.4202 & 0.9493 & -10.7997 & -2.1349 \\ 34.8970 & 4.3753 & -42.9039 & -5.8960 \end{pmatrix},$$

$$B^* = \begin{pmatrix} 0.1521 & 0 & 0 & 0 \\ 0 & 0.1986 & 0 & 0 \\ 0 & 0 & 0.1736 & 0 \\ 0 & 0 & 0 & 0.2269 \end{pmatrix} \tag{5.1b}$$

in the Jacobi case, and

$$S = \begin{pmatrix} 0.2030 & 0.3803 & -0.0763 & -0.0904 \\ -0.9495 & -0.8908 & 0.1977 & 0.2080 \\ 0.0858 & 0.1003 & -0.1466 & -0.0182 \\ -0.2233 & -0.2275 & -0.9662 & -0.9738 \end{pmatrix},$$
$$B^* = \begin{pmatrix} 0.2269 & 0 & 0 & 0 \\ 0 & 0.1737 & 0 & 0 \\ 0 & 0 & 0.1986 & 0 \\ 0 & 0 & 0 & 0.1521 \end{pmatrix} \tag{5.2b}$$

in the Gauss–Seidel case.

Since this paper aims at a comparison of algorithmic properties, we avoided effects of stepsize and iteration strategies by performing the experiments with fixed stepsizes h and fixed numbers of outer iterations m and inner iterations r. Furthermore, the Jacobian and the LU-decompositions were computed in each integration step.

The tables of results list for various values of the numbers of outer iterations m and inner iterations r the minimal number of correct digits at the end point:

$$\mathrm{cd} := -\log_{10} \left\| \boldsymbol{y}_{\mathrm{end}} - \boldsymbol{y}(t_{\mathrm{end}}) \right\|_\infty. \tag{5.3}$$

Here, $\boldsymbol{y}_{\mathrm{end}}$ denotes the numerical solution at the end point t_{end}. Negative cd values will be denoted by cd = –.

5.1. *The transistor amplifier (index 1)*

The first test problem is the transistor amplifier given in [6] on the interval $[0, 0.2]$ (see also [10]). This nonlinear, eight-dimensional problem of index 1 can be represented in the implicit form

$$K\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y})$$

with a constant, singular capacity matrix K. Table 5 lists results for the EPL predictor and $h = 2 \times 10^{-4}$. In both versions, only two inner iterations are needed to produce the same accuracy as the modified Newton process, but taking just one inner iteration seems to be the most efficient strategy. Furthermore,

Table 5
Transistor amplifier with EPL predictor and $h = 2 \times 10^{-4}$

	Jacobi version			Newton		Gauss–Seidel version	
m	$r = 1$	$r = 2$	→	$r = \infty$	←	$r = 2$	$r = 1$
1	–	4.6	→	4.6	←	4.6	–
2	6.5	6.6	→	6.6	←	6.6	–
3	7.7	7.5	→	7.5	←	7.5	7.0
4	8.1	8.0	→	8.0	←	8.0	7.2
⋮	⋮	⋮		⋮		⋮	⋮
∞	9.7	9.7	→	9.7	←	9.7	9.7

Table 6
Transistor amplifier with LSV predictor and $h = 2 \times 10^{-4}$

	Jacobi version					Newton		Gauss–Seidel version			
m	$r=1$	$r=2$	$r=3$	$r=4$	→	$r=\infty$	←	$r=4$	$r=3$	$r=2$	$r=1$
1	–	2.1	2.9	3.1	→	3.2	←	3.1	2.8	2.0	1.1
2	1.4	3.7	4.7	4.4	→	4.4	←	4.4	4.2	3.7	2.0
3	2.5	4.9	5.9	5.8	→	5.8	←	5.8	5.6	4.9	2.7
4	3.4	6.0	6.6	6.7	→	6.7	←	6.8	6.9	6.2	3.7
⋮	⋮	⋮	⋮	⋮		⋮		⋮	⋮	⋮	⋮
∞	9.7	9.7	9.7	9.7	→	9.7	←	9.7	9.7	9.7	9.7

Table 7
The car axis problem with EPL predictor and $h = 3 \times 10^{-3}$

	Jacobi version					Newton		Gauss–Seidel version			
m	$r=1$	$r=2$	$r=3$	$r=4$	→	$r=\infty$	←	$r=4$	$r=3$	$r=2$	$r=1$
1	–	–	0.8	2.5	→	2.5	←	2.5	0.8	–	–
2	1.8	2.0	2.4	5.4	→	5.4	←	5.4	2.4	1.9	–
3	1.9	5.3	6.5	6.5	→	6.6	←	6.6	5.4	4.3	1.8
4	2.4	6.3	6.6	6.6	→	6.6	←	6.6	6.6	6.0	2.6
⋮	⋮	⋮	⋮	⋮		⋮		⋮	⋮	⋮	⋮
∞	6.6	6.6	6.6	6.6	→	6.6	←	6.6	6.6	6.6	6.6

in accordance with Table 3, the Jacobi version performs better than the Gauss–Seidel version (note that the outer iteration process converges relatively slowly).

Next, we apply the LSV predictor. According to Table 3, now Gauss–Seidel should be the superior one. Table 6 shows that Gauss–Seidel does perform slightly better than Jacobi. Furthermore, a comparison with Table 5 reveals that the EPL predictor is considerably more efficient than the LSV predictor because of its higher order. We also tested the GLM predictor, but it could not beat the EPL predictor. Apparently, a higher order of accuracy is more important than smaller amplification factors.

5.2. *The car axis problem (index 3)*

Table 6 presents results for the more complicated index 3 car axis problem consisting of 10 DAEs [10]. As in Table 5, Jacobi is slightly better than Gauss–Seidel and a one-inner-iteration strategy is most efficient (note that here the outer iteration process converges relatively fast).

5.3. *Concluding remarks*

From the results in Tables 5–7 we may draw the following conclusions:

(i) The PILSRK inner iteration process profits most from high-order predictors.

(ii) If higher-order predictors like EPL are used, then the Gauss–Seidel version $\widetilde{C} \neq O$ converges slightly slower than the Jacobi version $\widetilde{C} = O$.

(iii) If the number of outer iterations m increases, then the number of inner iterations r can be chosen smaller.

(iv) In a (fixed m, fixed r) strategy, the one-inner-iteration strategy together with a high-order predictor seems to be most efficient. A dynamic iteration strategy is expected to perform several inner iterations in the first few outer iterations and only one inner iteration in the later outer iterations.

References

[1] C. Bendtsen, Highly stable parallel Runge–Kutta methods, *Appl. Numer. Math.* 21 (1996) 1–8.

[2] K.E. Brenan, S.L. Campbell and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential–Algebraic Equations* (North-Holland, New York, 1989).

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

[4] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations* II. *Stiff and Differential–Algebraic Problems* (Springer, Berlin, 1996).

[5] E. Hairer, RADAUP (1996), available via WWW at URL: ftp://ftp.unige.ch/pub/doc/math/stiff/radaup.f.

[6] E. Hairer, C. Lubich and M. Roche, *The Numerical Solution of Differential–Algebraic Systems by Runge–Kutta Methods*, Lecture Notes in Mathematics 1409 (Springer, Berlin, 1989).

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

[8] P.J. van der Houwen and J.J.B. de Swart, Triangularly implicit iteration methods for ODE-IVP solvers, *SIAM J. Sci. Comput.* 18 (1996) 41–55.

[9] P.J. van der Houwen and J.J.B. de Swart, Parallel linear system solvers for Runge–Kutta methods, *Adv. Comput. Math.* 7 (1997) 157–181.

[10] W.M. Lioen, J.J.B. de Swart and W.A. van der Veen, Test set for IVP solvers (1996), available via WWW at URL: http://www.cwi.nl/cwi/projects/IVPtestset.shtml.

[11] E. Messina, J.J.B. de Swart and W.A. van der Veen, Parallel iterative linear solvers for multistep Runge–Kutta methods, Preprint NM-R9619, CWI, Amsterdam (1996).

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