Parallel Linear System Solvers for Runge-Kutta Methods

P.J. van der Houwen & J.J.B. de Swart
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract

If the nonlinear systems arising in implicit Runge-Kutta methods like the Radau IIA methods are iterated by (modified) Newton, then we have to solve linear systems whose matrix of coefficients is of the form $I - A\otimes hJ$ with A the Runge-Kutta matrix and J an approximation to the Jacobian of the righthand side function of the system of differential equations. For larger systems of differential equations, the solution of these linear systems by a direct linear solver is very costly, mainly because of the LU-decompositions. We try to reduce these costs by solving the linear systems by a second (inner) iteration process. This inner iteration process is such that each inner iteration again requires the solution of a linear system. However, the matrix of coefficients in these new linear systems is of the form $I - B\otimes hJ$ where B is similar to a diagonal matrix with positive diagonal entries. Hence, after performing a similarity transformation, the linear systems are decoupled into s subsystems, so that the costs of the LU-decomposition are reduced to the costs of s LU-decompositions of dimension d. Since these LU-decompositions can be computed in parallel, the effective LU-costs on a parallel computer system are reduced by a factor s^3. It will be shown that matrices B can be constructed such that the inner iterations converge whenever A and J have their eigenvalues in the positive and nonpositive halfplane, respectively. The theoretical results will be illustrated by a few numerical examples. A parallel implementation on the four-processor Cray-C98 / 4256 shows a speed-up ranging from at least 2.4 until at least 3.1 with respect to RADAU5 applied in one-processor mode.

CR Subject Classification (1991): G.1.7
Keywords and Phrases: numerical analysis, convergence of iteration methods, Runge-Kutta methods, parallelism.

Note: The research reported in this paper was partly supported by the Technology Foundation (STW) in the Netherlands.

1. Introduction

Suppose that we integrate the IVP

$$(1.1)\quad \frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}),\quad \mathbf{y}(t_0) = \mathbf{y}_0,\quad \mathbf{y}, \mathbf{f} \in R^d$$

by an implicit step-by-step method. In general, this requires in each step the solution of a nonlinear system of the form

$$(1.2)\quad \mathbf{R}(\mathbf{Y}_n) = \mathbf{0},\quad \mathbf{R}(\mathbf{Y}) := \mathbf{Y} - h(A\otimes I)\mathbf{F}(\mathbf{Y}) - \mathbf{W}_{n-1},$$

where A denotes an s-by-s matrix (assumed to be nondefective), I is the d-by-d identity matrix, $\mathbf{W}_{n-1}$ contains information from preceding steps, h is the stepsize $t_n - t_{n-1}$, and $\otimes$ denotes the Kronecker product. It will always be assumed that A is nondefective and has its eigenvalues in the positive halfplane. The s components $\mathbf{Y}_{ni}$ of the sd-dimensional solution vector $\mathbf{Y}_n$ represent s numerical approximations to the s exact solution vectors $\mathbf{y}(t_{n-1} + c_i h)$; here, $\mathbf{c} = (c_i)$ denotes the abscissa vector whose components c_i are assumed distinct. Furthermore, for any vector $\mathbf{Y}_n = \big(\mathbf{Y}_{ni}\big)$, $\mathbf{F}(\mathbf{Y}_n)$ contains the derivative values $\big(\mathbf{f}(\mathbf{Y}_{ni})\big)$. In the following, we shall use the notation I for any identity matrix. However, its order will always be clear from the context. The solution $\mathbf{Y}_n$ of (1.2) will be called the *stage vector*, $\mathbf{y}_n$ the *step point value*, s the *number of stages*, and A the *Runge-Kutta matrix*.

Usually, the nonlinear system (1.2) is solved by modified Newton iteration. This leads to linear systems whose matrix of coefficients is of the form $I - A \otimes hJ$ with J an approximation to the Jacobian of the righthand side function **f**. The solution of these systems may be extremely costly. For example, if a direct solver is used, then in general the LU-decomposition requires $\frac{2}{3}$ $s^3 d^3$ arithmetic operations which is considerable, even for moderate values of d (say $d \approx 10$). Moreover, there is only a limited intrinsic parallelism in building the LU decomposition of the matrix $I - A \otimes hJ$.

1.1. Reduction of computational costs

We briefly survey various approaches to reduce the computational costs associated with the solution of the Newton systems using parallel computer systems. Firstly, one may look for special methods in which A is a triangular matrix with positive diagonal entries like the DIRK type methods. Then, confining our considerations to the costs of the LU decomposition, we see that the effective LU-costs on s processors reduce to $\frac{2}{3}$ d^3 operations, a factor s^3 less than those needed for the Newton process. However, these DIRK type methods also have disadvantages. In the case of one-step DIRKs available in the literature, the step point order is at most 4 and they have a relatively low stage order which may be a disadvantage in certain classes of stiff IVPs. Higher step point orders and stage orders can be obtained in the class of multistep RK methods (cf. Burrage and Chipman [3]), but they have the disadvantage of quite large abscissae values c_i (much larger than 1).

More sophisticated than the DIRK methods are methods characterized by matrices A with only positive eigenvalues such as the one-step RK methods of Nørsett [16], Burrage [1] and Orel [17]. By performing a similarity transformation (or Butcher transformation [5]), the linear systems can be decoupled into s subsystems of dimension d. Again, the effective LU-costs reduce by a factor s^3, and moreover, the stage order and step point order are much higher than for DIRK methods. However, a possible disadvantage of these methods is the lack of superconvergence at the step points.

Finally, one may choose the classical RK methods possessing both a high stage order and a high step point order, but also one or more complex eigenvalues. Again, applying a similarity transformation, the Newton system is transformed to block-diagonal form with (real) diagonal blocks, each block corresponding to an eigenvalue of A. If an eigenvalue is real, then the associated diagonal block is of order d, otherwise it has order 2d. The LU-costs of these blocks are reduced to $\frac{2}{3}$ d^3 and $\frac{16}{3}$ d^3 operations, so that effectively the LU-costs are $\frac{16}{3}$ d^3 operations, irrespective the value of s (the code RADAU5 of Hairer and Wanner [7] uses such a transformation).

1.2. Iterative solution of the linear systems

Instead of using direct solution methods, one may also look for *iterative* linear solvers, such as GMRES or preconditioned GMRES (see e.g. Burrage [2] where further references are given).

In this paper, we shall follow an approach that is a mixture of an iterative and a direct approach. It allows A to have complex eigenvalues (in the positive halfplane), so that the superconvergent RK methods like the Radau IIA methods are included. The linear systems arising in the modified Newton method are solved by an iterative method (the *inner iteration process*), which needs itself LU decompositions of matrices, but these matrices are only of dimension d. In fact, the linear systems to be solved have a matrix of coefficients of the form $I - B \otimes hJ$ where B is similar to a diagonal matrix with positive diagonal entries. Hence, after performing a similarity transformation, the effective LU costs are $\frac{2}{3}$ d^3 operations like the methods of Burrage and Orel. We shall refer to this inner iteration process by PILSRK (Parallel Iterative Linear System solver for RK methods). The combination of the modified Newton and the PILSRK method will be called the Newton-PILSRK method.

There are several options for choosing the matrix B. The most simple approach chooses $B = D$ where D is a *diagonal* matrix (with positive entries), so that the sd-dimensional system can directly be split into s uncoupled subsystems of dimension d which can be solved concurrently. In fact, we can employ the same matrices D as used in the Parallel Diagonal-implicitly Iterated RK methods (PDIRK methods) analysed in [10]. The PDIRK method is also an iterative method, but unlike the PILSRK method it is a *nonlinear* system solver and directly iterates on the nonlinear system (1.2). Using results derived by Lioen [13] for PDIRK matrices, it can be shown that for the first eight Radau IIA correctors, the PILSRK methods are A-convergent, that is, it converges if J has its eigenvalues in the nonpositive halfplane. Furthermore, these PDIRK matrices have the property that the stiff components are removed from the iteration error within s iterations. However, a disadvantage of the PDIRK matrices is the poor convergence (or even divergence) of the PILSRK method in the first few iterations which is worse as the number of stages of the underlying RK corrector increases. Such a convergence behaviour is highly undesirable if we want to apply step-parallel iteration, where the iteration process is already started at the next step point t_{n+1} before the iterates at t_n have converged. A poor initial convergence implies that no accurate predictor value is available for starting the iteration process at t_{n+1}. A substantial improvement in the initial phase of the convergence of the PILSRK method is obtained by employing the matrices L used in the Parallel Triangular-implicitly Iterated RK methods (PTIRK methods) constructed in [11] (like the PDIRK methods, the PTIRK methods are *nonlinear* system solvers). The PTIRK matrices L are defined by the lower triangular factor of the Crout decomposition LU of the RK matrix A. By virtue of results obtained by Hoffmann and De Swart [8], it can be shown that for all RK correctors that are based on collocation with positive, distinct abscissae, the matrix L has positive diagonal entries and that the PILSRK method is A-convergent. Furthermore, like the PDIRK matrices, the PTIRK matrices have the property that the stiff components are removed from the iteration error within s iterations. After performing a similarity transformation, the effective LU-costs are reduced by a factor s^3. A preliminary parallel implementation of the Newton-PILSRK method based on the one-step 4-stage Radau IIA formula and using the PTIRK matrix showed on the four-processor Cray-C98 / 4256 speed-up factors ranging from at least 2.4 until at least 3.1 with respect to RADAU5 in one-processor mode (cf. [11]).

1.3. Outline of the paper

The aim of the present paper is to find matrices B that are still more effective than the PTIRK matrices L. Our starting point is the representation B = QTQ^{-1} with T a lower triangular matrix with positive diagonal entries and with Q a nonsingular transformation matrix such that $Q^{-1}AQ$ is lower *block-triangular*. It will be shown that matrices T and Q exist such that:

(i) B is nondefective and has positive eigenvalues,
(ii) the PILSRK method is A-convergent whenever A has its eigenvalues in the positive halfplane,
(iii) the stiff components are removed from the iteration error in the second iteration,
(iv) the spectral radius of the iteration-error-amplification matrix is minimized in the left halfplane.

The difficult part is the construction of matrices Q such that the iteration-error-amplification matrix has a sufficiently small norm. In this paper, we construct transformation matrices so that $Q^{-1}AQ$ is *block-diagonal* (in a forthcoming paper, we shall deal with alternative families of transformation matrices). For the 4-stage and 8-stage Radau IIA correctors, matrices Q will be constructed such that the Euclidean norm of powers of the iteration-error-amplification matrix are satisfactorily small.

As soon as T and Q, and hence B, are obtained, we can compute the diagonalizing similarity transformation, to obtain a highly parallel linear system solver.

In this paper, we have restricted our analysis of the Newton-PILSRK method to the case where (1.2) represents the class of one-step Radau IIA methods, that is, A is the Radau IIA matrix and $\mathbf{W}_{n-1} := (E\otimes I)\mathbf{Y}_{n-1}$ with E = $(\mathbf{0}, \ldots, \mathbf{0}, \mathbf{e})$, $\mathbf{e}$ being an s-dimensional vector with unit entries. These methods are of particular interest because of their high step point order p = 2s-1 and high stage order q = s, their stiff accuracy and their excellent stability properties. The Newton-PILSRK methods were applied to a few problems taken from the literature. The results show a considerable improvement of the convergence in the first few outer iterations. Recalling that a parallel implementation of the Newton-PILSRK method using the PTIRK matrices already shows a speed-up factor of at least 2.4 with respect to RADAU5, we expect that using the new matrices B = QTQ^{-1} will yield a further speed-up. The parallel implementation of the new methods will be subject of future research.

Finally, we remark that it may well be that the class of multistep RK methods of Radau type (cf. Hairer and Wanner [7, p. 293]) is a better choice for the corrector equation (1.2) than the one-step Radau methods. For nonstiff IVPs, Burrage and Suhartanto [4] have investigated the use of parallel iteration methods for such correctors and they report promising results. This indicates that applying the PILSRK approach of this paper to the Newton systems arising in multistep Radau methods may lead to quite effective parallel IVP methods.

2. The parallel iterative linear system solver

Consider the modified Newton iteration scheme for solving the corrector equation (1.2):

$$(2.1)\quad \big(I - A\otimes hJ\big)\big(\mathbf{Y}^{(j)} - \mathbf{Y}^{(j-1)}\big) = -\,\mathbf{R}\big(\mathbf{Y}^{(j-1)}\big), \quad j = 1, 2, \ldots, m,$$

where $J = \partial\mathbf{f}/\partial\mathbf{y}$ is evaluated at t_{n-1}, $\mathbf{Y}^{(0)}$ is the initial iterate to be provided by some predictor formula, and where $\mathbf{Y}^{(m)}$ is adopted as the solution $\mathbf{Y}_n$ of the corrector equation (1.2). Each iteration with (2.1) requires the solution of an sd-dimensional linear system for the Newton correction $\mathbf{Y}^{(j)} - \mathbf{Y}^{(j-1)}$. As already observed, direct solution of this Newton system can be extremely costly and transformation to block-diagonal form reduces computational costs considerably. In order to achieve a still greater reduction of the computational complexity we follow an alternative approach by applying an iterative linear solver to the Newton systems in (2.1). This solver again requires the solution of linear systems, but these systems are only of dimension d. It is tuned to the RK structure of the systems in (2.1) and possesses a lot of intrinsic parallelism. This Parallel Iterative Linear System solver for RK methods (PILSRK method) is defined by

$$(2.2)\quad \begin{aligned} &\big(I - B\otimes hJ\big)\big(\mathbf{Y}^{(j,\nu)} - \mathbf{Y}^{(j,\nu-1)}\big) = -\,\big(I - A\otimes hJ\big)\mathbf{Y}^{(j,\nu-1)} + \mathbf{C}^{(j-1)}, \\ &\mathbf{C}^{(j-1)} := \big(I - A\otimes hJ\big)\mathbf{Y}^{(j-1)} - \mathbf{R}\big(\mathbf{Y}^{(j-1)}\big), \end{aligned} \qquad \nu = 1, 2, \ldots, r,$$

where $\mathbf{Y}^{(j,0)} = \mathbf{Y}^{(j-1,r)}$ and where $\mathbf{Y}^{(m,r)}$ is accepted as the solution $\mathbf{Y}_n$ of the corrector equation (1.2). The matrix B is assumed to be nondefective and to have positive eigenvalues. Note that $\mathbf{C}^{(j-1)}$ does not depend on ν, so that the application of the inner iteration process requires only one evaluation of the function $\mathbf{R}$. The processes (2.1) and (2.2) may be considered as the *outer* and *inner* iteration processes.

In order to construct a suitable matrix B, we observe that the condition on the spectrum of B implies that we can write $B = QTQ^{-1}$ with Q an arbitrary real, nonsingular matrix and T a lower triangular matrix with positive diagonal entries. Hence, by performing the transformation $\mathbf{Y}^{(j,\nu)} = (Q\otimes I)\tilde{\mathbf{Y}}^{(j,\nu)}$, we obtain

$$(2.3)\quad \big(I - T\otimes hJ\big)\big(\tilde{\mathbf{Y}}^{(j,\nu)} - \tilde{\mathbf{Y}}^{(j,\nu-1)}\big) = -\,\big(I - \tilde{A}\otimes hJ\big)\tilde{\mathbf{Y}}^{(j,\nu-1)} + (Q^{-1}\otimes I)\mathbf{C}^{(j-1)}, \quad \nu = 1, 2, \ldots, r,$$

where $\tilde{A} = Q^{-1}AQ$ and $\tilde{\mathbf{Y}}^{(j,0)} = (Q^{-1}\otimes I)\mathbf{Y}^{(j-1)}$. If for a given j, the transformed inner iterates $\tilde{\mathbf{Y}}^{(j,\nu)}$ converge to a vector $\tilde{\mathbf{Y}}^{(j,\infty)}$, then the Newton iterate defined by (2.1) can be obtained from $\mathbf{Y}^{(j)} = (Q\otimes I)\tilde{\mathbf{Y}}^{(j,\infty)}$. Given the matrix A, the PILSRK method (2.3) is completely defined by the matrix pair (T,Q) and will be denoted by PILSRK(T,Q). The representation (2.3) will be the starting point for the construction of the matrix B.

Before discussing the computational costs of the actual implementation of the Newton-PILSRK method {(2.1),(2.3)}, we should specify the matrix B. This will be the subject of Section 3. Details on the computational complexity can be found in Section 4.2.

Remark 2.1. In the first Newton iterations, it seems a waste to perform many inner iterations with the PILSRK method, because there is no point in computing a very accurate approximation to $\mathbf{Y}^{(j)}$, as long as $\mathbf{Y}^{(j)}$ is itself a poor approximation to $\mathbf{Y}_n$. Likewise, in later outer iterations, we expect that only a few inner iterations suffice to solve $\mathbf{Y}^{(j)}$ from (2.1). In the extreme case, only one inner iteration is performed in each outer iteration. In such an iteration strategy, the Newton-PILSRK iteration method {(2.1),(2.2)} simplifies to

$$(2.4) \qquad \big(I - B\otimes hJ\big)\big(\mathbf{Y}^{(j)} - \mathbf{Y}^{(j-1)}\big) = - \mathbf{R}\big(\mathbf{Y}^{(j-1)}\big), \quad j = 1, 2, \ldots , m.$$

However, this process may converge very slowly in the first few outer iterations, and it is recommended, either to use highly accurate predictor formulas for $\mathbf{Y}^{(0)}$ or to introduce a dynamic iteration strategy so that when necessary, sufficiently many inner iterations in the first few outer iterations are performed.

Notice also that the iterative method obtained from (2.1) by using a splitting of A into B and A-B is identical with the iteration method (2.4). ♦

3. Construction of the matrix B

Given the matrix A, the PILSRK method (2.2) is completely determined by the matrix $B = QTQ^{-1}$. In the construction of B, the region of convergence and the averaged amplification factors for the iteration errors play a central role.

3.1. Convergence region of the PILSRK method

In order to analyse the region of convergence for the PILSRK method, we consider the error recursion

$$(3.1) \ \mathbf{Y}^{(j,\nu)} - \mathbf{Y}^{(j)} = M\big(\mathbf{Y}^{(j,\nu-1)} - \mathbf{Y}^{(j)}\big), \quad M := \big(I - B\otimes hJ\big)^{-1} \big((A - B)\otimes hJ\big).$$

We have convergence if the powers M^ν of the amplification matrix M tend to zero as $\nu \to \infty$, that is, if the spectral radius $\rho(M)$ of M is less than 1. The eigenvalues of M are given by the eigenvalues of the matrix

$$(3.2) \ Z(z) := z \big(I - zB\big)^{-1} \big(A - B\big), \quad z := h\lambda,$$

where λ runs through the eigenvalues of J. We call $\Gamma := \{z\colon \rho\big(Z(z)\big) < 1\}$ the *region of convergence* of the PILSRK method. Thus, the method converges if the eigenvalues of hJ lie in Γ. If Γ contains the whole nonpositive halfplane, then the method will be called *A-convergent*.

We shall call Z(z) the *amplification matrix* at the point z and $\rho\big(Z(z)\big)$ the (*asymptotic*) *amplification factor* at z. The maximum of $\rho\big(Z(z)\big)$ in the left halfplane $\mathrm{Re}(z) \le 0$ will be denoted by ρ.

In [10] and [11] where the PDIRK and PTIRK methods were analysed, it turned out that strong damping of the *stiff* error components, that is, small amplification factors for error components corresponding to eigenvectors of J with eigenvalues λ of large magnitude, is crucial for a fast overall convergence. This leads us to require the matrix B to be such that $\rho(Z(\infty)) = \rho(I - B^{-1}A)$ vanishes. If we succeed in finding such matrices B, then $Z^s(\infty) = O$, so that within s iterations, the components corresponding to $|\lambda| = \infty$ are removed from the iteration error (this can be verified by considering the Schur decomposition of $Z^s(\infty)$).

As an example, let Q = I and let T be a diagonal matrix D, so that B = D. Lioen [13] showed that for the s-stage Radau IIA correctors with $s \le 8$, it is possible to construct diagonal matrices D satisfying $\rho(I - D^{-1}A) = 0$ such that the generated PILSRK(D,I) method is A-convergent. These matrices are also used in the PDIRK methods studied in [10], and will therefore be called PDIRK matrices.

The next theorem defines a family of PILSRK(T,Q) methods automatically satisfying the condition $\rho(I - B^{-1}A) = 0$.

Theorem 3.1. Let Q be an arbitrary, nonsingular matrix and let $B = QTQ^{-1}$, where T is the lower triangular factor in the Crout-decomposition of $\tilde{A} := Q^{-1}AQ$. Then, the asymptotic amplification factor vanishes at infinity.

Proof. Let TU represent the Crout-decomposition of $\tilde{A}$. Then $Q^{-1}Z(\infty)Q = I - Q^{-1}B^{-1}AQ = I - T^{-1}\tilde{A} = I - U$ is strictly upper triangular. Hence, $\rho(Q^{-1}Z(\infty)Q) = \rho(Z(\infty)) = 0$. ♦

The matrix B in the PILSRK methods characterized by this theorem does not necessarily have positive eigenvalues and hence, does not automatically generate A-convergent methods. This requires special transformation matrices Q. Let us again consider the case where Q = I. Then, B equals the lower triangular factor in the Crout-decomposition of A, that is, B equals the PTIRK matrix L derived in [11]. In [8], Hoffmann and De Swart were able to prove that the PTIRK matrix L possesses positive diagonal entries for all collocation-based RK correctors with positive, distinct abscissas, so that B has positive eigenvalues as required. Furthermore, numerical computations in [11] showed the A-convergence for a large number of RK correctors based on Gaussian quadrature formulas.

The aim of this paper is to derive A-convergent methods with $\rho(I - B^{-1}A) = 0$ for more general pairs (T,Q) than the PTIRK pair (L,I), and to find pairs (T,Q) such that we can a priori prove both the positiveness of the eigenvalues of B and the A-convergence of the generated iteration method.

Let us choose Q such that $\tilde{A} := Q^{-1}AQ = (\tilde{A}_{kl})$ is a (real) σ-by-σ lower block-triangular matrix, of which the diagonal blocks $\tilde{A}_{kk}$ are either one-by-one or two-by-two matrices. If ξ_k is a real eigenvalue of A, then $\tilde{A}_{kk} = \xi_k$, and if $\xi_k \pm i\eta_k$ is a complex eigenvalue pair of A, then

$$(3.3)\quad \tilde{A}_{kk} = \begin{pmatrix} a_k & b_k \\ c_k & 2\xi_k - a_k \end{pmatrix}, \quad b_k = -c_k^{-1}(a_k^2 - 2\xi_k a_k + \alpha_k^2), \quad c_k \ne 0, \quad \alpha_k := \sqrt{\xi_k^2 + \eta_k^2}\,,$$

where a_k and c_k are free parameters. In the following, K will denote the set of integers with the property that $\eta_k \ne 0$ whenever $k \in K$.

A natural choice for T now is

$$\text{(3.4a)}\quad T := \begin{pmatrix} T_{11} & O & O & O & \dots \\ \tilde{A}_{21} & T_{22} & O & O & \dots \\ \tilde{A}_{31} & \tilde{A}_{32} & T_{33} & O & \dots \\ . & . & . & . & . \end{pmatrix}, \; T_{kk} := \begin{pmatrix} u_k & 0 \\ v_k & w_k \end{pmatrix} \text{ if } k \in K, \quad T_{kk} = \xi_k \text{ otherwise,}$$

where u_k, v_k and w_k are free parameters with u_k and w_k assumed to be positive.

Theorem 3.2. Let A have its eigenvalues in the positive halfplane, let $\tilde{A} := Q^{-1}AQ = (\tilde{A}_{kl})$ be lower block-triangular, let the diagonal blocks be defined by (3.3) and let $T = T(\gamma)$ be defined by (3.4a) with

$$\text{(3.4b)}\quad u_k = \gamma\alpha_k, \quad v_k = c_k\alpha_k\frac{(\gamma^2 - 1)a_k - 2\gamma^2\xi_k + 2\gamma\alpha_k}{\gamma(a_k{}^2 - 2\xi_k a_k + \alpha_k{}^2)}, \quad w_k = \frac{\alpha_k}{\gamma},$$

where γ is a positive parameter. Then, for all a_k and c_k the following assertions hold:

(i) The asymptotic amplification factor vanishes at infinity.
(ii) B has positive eigenvalues and if $\gamma \neq 1$ it is nondefective.
(iii) The PILSRK(T(γ),Q) method is A-convergent with $\rho = \max\{|1 - 2\gamma(\gamma^2 + 1)^{-1}\xi_k\alpha_k{}^{-1}|: k \in K\}$.

Proof. Let

$$\text{(3.5)}\quad \tilde{Z}(z) := Q^{-1}Z(z)Q = z\,(I - zT)^{-1}\,(\tilde{A} - T), \quad z := h\lambda.$$

If T is of the form (3.4a), then the value of $\rho(Z(z)) = \rho(\tilde{Z}(z))$ equals the maximum of the spectral radius $\rho(\tilde{Z}_{kk}(z))$ of the diagonal blocks

$$\text{(3.6)}\quad \tilde{Z}_{kk} := z\,(I - zT_{kk})^{-1}\,(\tilde{A}_{kk} - T_{kk})$$

of $\tilde{Z}$. Here, $\tilde{Z}_{kk}$ vanishes if the underlying eigenvalue of A is real. Hence, in order to achieve $\rho(Z(\infty)) = 0$, we choose the T_{kk} with $k \in K$ such that the spectral radius of the corresponding diagonal blocks $Z_{kk}(z)$ vanishes at infinity.
We derive from (3.3) and (3.6) that the eigenvalues ζ_k of $\tilde{Z}_{kk}$ satisfy the characteristic equation

$$\text{(3.7)}\quad \det\begin{pmatrix} (a_k\text{-}u_k)z - \zeta_k(1\text{-}zu_k) & b_k z \\ (c_k\text{-}v_k)z + \zeta_k v_k z & (2\xi_k\text{-}a_k\text{-}w_k)z - \zeta_k(1\text{-}zw_k) \end{pmatrix} = 0.$$

It is easily verified that $\zeta_k = \zeta_k(z)$ vanishes at infinity if u_k, v_k and w_k are defined according to (3.4b). Hence, $\rho(\tilde{Z}_{kk}(z))$ vanishes at infinity which proves part (i) of the theorem.
Since the eigenvalues of B are given by (u_k, w_k) for $k \in K$ and by ξ_k for $k \notin K$, and because we assumed $\gamma > 0$, (3.4b) also implies that B has positive eigenvalues and if $\gamma \neq 1$ it is nondefective, proving part (ii).
The characteristic equation (3.7) is solved by

(3.8) $\zeta_k = 0, \quad \zeta_k = \dfrac{(2\xi_k - u_k - w_k)z}{(1 - zu_k)(1 - zw_k)}$,

so that $\rho\big(\tilde{Z}(z)\ \big)$ equals the maximum of the values $|\zeta_k(z)|$. Since $\zeta_k(z)$ is regular in left half plane (provided that u_k and w_k are positive), its maximum in the left halfplane $\mathrm{Re}(z) \le 0$, to be denoted by ρ_k, is assumed on the imaginary axis. It is easily verified that

(3.9) $\rho\big(\tilde{Z}_{kk}(iy)\big) = |\zeta_k(iy)| = \dfrac{|2\xi_k - u_k - w_k|\ |y|}{\sqrt{(1 + u_k^2y^2)(1 + w_k^2y^2)}}$

assumes an absolute maximum at $y = y_0 := (u_kw_k)^{-1/2}$ and that the maximum value ρ_k of $\rho\big(Z_{kk}(iy)\big)$ is given by $\rho_k = \big|1 - 2\xi_k(u_k + w_k)^{-1}\big| = \big|1 - 2\gamma(\gamma^2 + 1)^{-1}\xi_k\alpha_k^{-1}\big|$, which is less than 1 whenever $\gamma\xi_k > 0$. This proves part (iii) of the theorem. ♦

The asymptotic amplification factor ρ is minimized for $\gamma = 1$ and assumes the minimal value $\rho = \max\{1 - \xi_k\alpha_k^{-1}: k \in K\}$. However, then the matrices T_{kk} are defective (because $u_k = w_k$). Hence, T cannot be diagonalized, and although the effective LU-costs are still reduced by a factor s, the Newton-PILSRK(T(1),Q) method should be considered as a σ-processor method, rather than an s-processor method. Fortunately, the asymptotic amplification factor varies slowly with γ, so that we can remove the defectness of T at the cost of a slight increase of ρ. For example, for the method defined by (3.4) we find for $\gamma = \frac{7}{8}$,

(3.10) $\quad \rho = \max\{1 - \frac{112}{113}\,\xi_k\alpha_k^{-1}: k \in K\}$,

which is only slightly larger than the minimal value. For a detailed discussion of the computational complexity of an implementation of the Newton-PILSRK(T(γ),Q) method, we refer to Section 4.2.

Remark 3.1. When faced with the problem of choosing a matrix T such that the eigenvalues of the matrix Z(z) are of small magnitude, it is tempting to minimize the magnitude of the matrix factor $\tilde{A}$ - T occurring in the matrix $\tilde{Z}(z)$ defined by (3.5). Since

$$\tilde{A} - T = \mathrm{diag}\big(\tilde{A}_{11} - T_{11}, ..., \tilde{A}_{\sigma\sigma} - T_{\sigma\sigma}\big),\ \tilde{A}_{kk} - T_{kk} = \begin{pmatrix} a_k\text{-}u_k & b_k \\ c_k\text{-}v_k & 2\xi_k\text{-}a_k\text{-}w_k \end{pmatrix}$$

and because for given a_k, the magnitude of the entry $b_k = -\,c_k^{-1}\big(a_k^2 - 2\xi_ka_k + \alpha_k^2\big)$ can be made as small as we want, we are led to zero the other three entries of $\tilde{A}_{kk} - T_{kk}$ by setting $u_k = a_k$, $v_k = c_k$ and $w_k = 2\xi_k - a_k$. This still leaves a_k as a free parameter which can be used to minimize b_k for given c_k, to obtain $a_k = \xi_k$ and $b_k = -\eta_k^2c_k^{-1}$. However, substitution of the parameters u_k, v_k, w_k, a_k and b_k into the characteristic equation (3.7) reveals that the nonzero eigenvalue is given by $\zeta_k = (\xi_k^2 - \alpha_k^2)z^2(1 - z\xi_k)^{-2}$, which assumes the extreme value $-\big(\eta_k\xi_k^{-1}\big)^2$ at infinity. Thus, we have no A-convergence when A has eigenvalues whose imaginary part exceeds its real part. Since many

RK methods based on Gaussian quadrature do have imaginary parts that exceed the real parts, the approach of minimizing the magnitude of $\tilde{A}$ - T is the wrong way to go. ♦

Remark 3.2. The family of matrices T defined by (3.4) contains the special case where T is defined by the lower triangular factor in the Crout-decomposition of $\tilde{A} := Q^{-1}AQ$ (see Theorem 3.1):

$$(3.11)\quad T_{kk} := \begin{pmatrix} a_k & 0 \\ c_k & \alpha_k^2 a_k^{-1} \end{pmatrix} \quad \text{if} \quad k \in K, \quad T_{kk} = \xi_k \ \text{ otherwise.}$$

This expression is also obtained from (3.4) by setting $\gamma = a_k\alpha_k^{-1}$. ♦

We conclude this section with listing values of ρ_k for a few Radau IIA correctors and for the iteration strategy PILSRK(T(7/8),Q) defined by Theorem 3.2. In addition, we list the values of ρ for PILSRK(D,I) with the PDIRK matrix D and for PILSRK(L,I) with the PTIRK matrix L. The figures in Table 3.1 show that on the basis of the asymptotic amplification factors, the PILSRK(T(7/8),Q) approach is superior to PILSRK(D,I) and PILSRK(L,I).

Table 3.1. Values of ρ_k for Radau IIA methods.

Iteration	k	s = 2	s = 3	s = 4	s = 6	s = 8
PILSRK(D,I)	-	0.26	0.40	0.52	0.72	0.90
PILSRK(L,I)	-	0.18	0.37	0.51	0.70	0.86
PILSRK(T(7/8),Q)	1	0.19	0.35	0.45	0.57	0.64
	2			0.06	0.21	0.33
	3				0.03	0.12
	4					0.02

3.2. Averaged amplification factors

Because the matrix M in (3.1) is not expected to be a normal matrix, the asymptotic amplification factor ρ discussed in the preceding section only gives an indication of the speed of convergence after a quite large number of iterations and does not give insight into the convergence behaviour in the initial phase of the iteration process. In fact, for large ν we have the estimate $\| M^\nu \| \le \kappa(S)\ [\rho(M)]^\nu$, where S represents the eigensystem of M, $\kappa(S) := \|S\|\ \|S^{-1}\|$ is the condition number of S, and where we assumed that M has eigenvalues of multiplicity 1 (cf. Varga [20]). In order to analyse the convergence rate in the first few iterations, one may use to the pseudo-eigenvalue analysis of Trefethen (see e.g. [18]). Alternatively, we may resort to a well-known theorem of Von Neumann. We shall follow the latter approach.

Let the logarithmic matrix norm $\mu[S]$ associated with the Euclidean norm be defined by $\mu[S] = \frac{1}{2}\ \lambda_{max}(S+S^H)$, where S^H is the complex transposed of S and $\lambda_{max}(.)$ denotes the algebraically largest eigenvalue. Then, we have:

Theorem 3.3. If $\mu[J] \le 0$, then $\| M^{\nu} \| \le \max\{\| Z^{\nu}(z)\| : \mathrm{Re}(z) \le 0\}$.

Proof. The proof is based on a generalization of a theorem of Von Neumann. Von Neumann's theorem states that, given a matrix J and a rational function R of z which has a bounded maximum norm $\| R \|_{\infty}$ in the left halfplane, then $\|R(J)\| \le \| R \|_{\infty}$, provided that $\mu[J] \le 0$ (see e.g. [7, p.179]). A matrix-valued version of Von Neumann's theorem, applying to the case where R(z) is a matrix with entries that are rational functions of z, was proved by Nevanlinna [15] (see also [7, p.356]). Since M^{ν} can be considered as a matrix-valued function of J (see (3.1)), we apply the matrix-valued version of Von Neumann's theorem with $R(z) := M^{\nu}(z)$, where

$$(3.12) \quad M^{\nu}(z) = \Big[\big(I - B\otimes zI\big)^{-1}\big((A\ - B)\otimes zI\big)\Big]^{\nu} = Z^{\nu}(z)\otimes I, \quad z = h\lambda.$$

This leads to the assertion of the theorem. ♦

This theorem motivates us to define the *local averaged amplification factor* at the point $z = h\lambda$ and the *global averaged amplification factor* by

$$(3.13a) \quad \rho^{(\nu)}(z) := \sqrt[\nu]{\| Z^{\nu}(z)\|} \ , \quad \rho^{(\nu)} := \max\{\rho^{(\nu)}(z) : \mathrm{Re}(z) \le 0\}.$$

Note that $\rho^{(\nu)}(z)$ approximates the asymptotic amplification factor $\rho(Z(z))$ as $\nu \to \infty$. Since in the left halfplane, $\rho^{(\nu)}(z)$ assumes its maximum on the imaginary axis, we may restrict our considerations to the imaginary axis, so that $\rho^{(\nu)} := \max\{\rho^{(\nu)}(iy) : y \ge 0\}$.
Theorem 3.3 indicates that we may expect faster convergence as $\rho^{(\nu)}$ is smaller. However, for small numbers of iterations (say $\nu \le 5$), $\rho^{(\nu)}$ will give a rather conservative estimate of the speed of convergence, because in some sense it is a 'worst case' estimate. In order to get insight into the amplification of *individual* error components, one may use the *local* amplification factor $\rho^{(\nu)}(z)$. Let us consider error components of the form $\mathbf{a} \otimes \mathbf{v}$, where $\mathbf{a}$ is an s-dimensional vector and $\mathbf{v}$ is an eigenvector of J with eigenvalue λ. By observing that $M^{\nu}(\mathbf{a} \otimes \mathbf{v}) = \big(Z^{\nu}(h\lambda)\otimes I\big)\, (\mathbf{a} \otimes \mathbf{v})$, it follows that $\rho^{(\nu)}(h\lambda)$ characterizes the averaged convergence of the error component corresponding with $h\lambda$ and that only for larger values of ν, when the error component with maximal $\rho^{(\nu)}(h\lambda)$ has become dominant, $\rho^{(\nu)}$ yields a quantitative estimate of the averaged convergence rate. In the first few iterations, when all error components play their part, the L_2 norm of the local amplification factor $\rho^{(\nu)}(z)$ provides more realistic estimates than the L_{∞} norm. This suggests to define a second global amplification factor:

$$(3.13b) \quad \sigma^{(\nu)} := \Big(\int_{y=0}^{y=\infty} [\rho^{(\nu)}(iy)]^2 \, dy \,)\Big)^{1/2}.$$

We did not succeed in finding an approach which really minimizes $\rho^{(\nu)}$. However, by considering the estimate

(3.14) $\| Z^{\nu}(z)\| = \| Q\tilde{Z}^{\nu}(z)Q^{-1}\| \leq \kappa(Q)\| \tilde{Z}^{\nu}(z)\|$,

we see that $\rho^{(\nu)}(z) \leq \left(\kappa(Q)\| \tilde{Z}^{\nu}(z) \|\right)^{1/\nu}$, which suggests the separate minimization of the factors $\kappa(Q)$ and $\| \tilde{Z}^{\nu}(z)\|$. We distinguish two approaches. In the first approach, we choose Q orthogonal, so that $\kappa(Q) = 1$. This can be achieved by defining $\tilde{A} := Q^{-1}AQ$ by the *real* Schur decomposition of A, leading to $a_k = \xi_k$ and $c_k = -\eta_k$ (see e.g. [6]). In a future paper, we shall elaborate on this case. In the present paper, we analyse a second approach where first $\| \tilde{Z}^{\nu}(z)\|$ is minimized and then $\kappa(Q)$. We shall do this for the case where $\tilde{A}$ is block-diagonal.

3.3. The block-diagonal case

In the remainder of this section, we shall analyse the case where $\tilde{A} := Q^{-1}AQ$ is block-diagonal and we shall use the still free parameters a_k and c_k for reducing the magnitude of $\| \tilde{Z}^{\nu}(z) \|$. However, we first justify our choice of a block-diagonal matrix $\tilde{A}$ by considering the damping of the stiff error components. The following theorem presents a result on the amplification of the stiff iteration errors.

Theorem 3.4. Let the conditions of Theorem 3.2 be satisfied and let $\tilde{A} := Q^{-1}AQ$ be block-diagonal. Then, the averaged amplification factor $\rho^{(\nu)}(z) = O\left(z^{(1-\nu)/\nu}\right)$ as $z \rightarrow \infty$ and the averaged global amplification factor $\sigma^{(\nu)}$ is finite if $\nu > 2$.

Proof. For $z \rightarrow \infty$, it follows from (3.2) that

$$Z(z) = \left(I - z^{-1}B^{-1}\right)^{-1} \left(I - B^{-1}A\right) = Z(\infty) + z^{-1}B^{-1}Z(\infty) + O(z^{-2}),\ \ Z(\infty) = I - B^{-1}A$$

(B may be assumed to be nonsingular because it is required to have positive eigenvalues). More generally, we have that

$$Z^{\nu}(z) = \sum_{i=1}^{\infty} (Z(\infty))^{\text{ceil}\,[\nu / i]}\, O\left(z^{1-i}\right),$$

where for any real x, ceil[x] denotes the first integer greater than or equal to x. We first show that all integer powers of $Z(\infty)$ greater than 1 vanish. Since $Z^{\nu} = Q\tilde{Z}^{\nu}Q^{-1}$, we have to show that all integer powers of $\tilde{Z}(\infty)$ greater than 1 vanish. Because $Q^{-1}AQ$ is block-diagonal, it follows from (3.4) that T is block-diagonal and from (3.5) that $\tilde{Z}(z)$ is block-diagonal. Hence, $\tilde{Z}(\infty)$ is block-diagonal with diagonal blocks $\tilde{Z}_{kk}(\infty)$. Since by virtue of Theorem 3.2, these blocks have a zero spectral radius, $(\tilde{Z}_{kk}(\infty))^{\nu}$ vanishes for $\nu \geq 2$ (this can easily be verified by considering their Schur decompositions). Consequently, $\tilde{Z}^{\nu}(\infty)$ itself, and hence $Z^{\nu}(\infty)$, vanishes for $\nu \geq 2$. From the expansion of $Z^{\nu}(z)$ we now immediately obtain $Z^{\nu}(z) = O\left(z^{1-\nu}\right)$ as $z \rightarrow \infty$. Substitution into (3.13) yields the result of the theorem. ♦

From this theorem it follows that the stiff error components may be considered as being removed from the iteration error within two (inner) iterations.

If we only know that $Z(\infty)$ has a zero spectral radius, as in the case of the PDIRK and PTIRK matrices D and L, then $Z^{\nu}(\infty)$ vanishes for $\nu \geq s$. Hence, by virtue of (3.14) it is seen that for $\nu \geq s$ we have $Z^{\nu}(z) = O\big(z^{1-\mathrm{ceil}\,[\nu/(s-1)]}\big)$ as $z \to \infty$, so that $\rho^{(\nu)}(z) = O\big(z^{(1-\mathrm{ceil}\,[\nu/(s-1)])/\nu}\big)$ as $z \to \infty$ and $\sigma^{(\nu)}$ is finite only if $2\big(1 - \mathrm{ceil}[\nu / (s-1)]\big) / \nu$ is less than -1, i.e. if $s \leq 2$. Thus, by virtue of the block-diagonality of the matrix $\tilde{A}$, the PILSRK(T,Q) has a much better stiff initial convergence than the PILSRK(D,I) and PILSRK(L,I) methods.

3.3.1. Reduction of $\|\tilde{Z}^{\nu}(z)\|$ in the left halfplane. We derive an estimate for the maximum norm of $\|Z^{\nu}(z)\|$ in the left halfplane by using the inequality (3.14). Since $\tilde{A} := Q^{-1}AQ$ is block-diagonal, $\tilde{Z}^{\nu}(z)$ is also block-diagonal with diagonal blocks $\tilde{Z}_{kk}^{\nu}(z)$ given by

$$(3.15)\qquad \tilde{Z}_{kk}(z) = \frac{z}{(1-u_kz)(1-w_kz)}\begin{pmatrix} (a_k-u_k)(1-w_kz) & b_k(1-w_kz) \\ (a_k-u_k)v_kz + (c_k-v_k)(1-u_kz) & b_kv_kz + (2\xi_k-a_k-w_k)(1-u_kz) \end{pmatrix}.$$

Here, the parameters u_k, v_k and w_k satisfy (3.4b). We first minimize the magnitude of $\|\tilde{Z}_{kk}^{\nu}(z)\|$. Note that this can be done independently of Q. Having found $\tilde{Z}_{kk}$, we determine Q by minimizing $\kappa(Q)$. The representation (3.15) suggests setting $a_k = u_k$ and $c_k = v_k$, to obtain for $k \in K$

$$(3.16)\qquad \tilde{Z}_{kk}^{\nu}(z) = \zeta_k^{\nu}(z)\begin{pmatrix} 0 & q_k(z) \\ 0 & 1 \end{pmatrix},\ \ \zeta_k(z) := \frac{(2\gamma\xi_k - \gamma^2\alpha_k - \alpha_k)z}{(1 - \gamma\alpha_kz)(\gamma - \alpha_kz)}\ ,\ \ q_k(z) := \alpha_k\frac{\gamma - \alpha_kz}{c_k}\ .$$

Note that setting $a_k = u_k$ in (3.4b) implies $c_k = v_k$.

Theorem 3.5. Let the conditions of Theorem 3.4 be satisfied, let $a_k = \gamma\alpha_k$, $|c_k| \geq \gamma^{-1}(1 + \gamma^2)\alpha_k$. Then, with respect to the maximum norm, the averaged amplification factor satisfies $\rho^{(\nu)} \leq [\kappa(Q)]^{1/\nu}\rho$, where $\rho = \max\{\big|1 - 2\gamma(\gamma^2 + 1)^{-1}\xi_k\alpha_k^{-1}\big|: k \in K\}$.

Proof. Let for any matrix M(z) depending on the complex variable z, $|||\, M \,|||$ denote the maximum norm of the function $\|M(z)\|$ in the left halfplane, where $\|\,.\,\|$ denotes the maximum matrix norm. It is easily seen that

$$(3.17)\qquad |||\,\tilde{Z}\,||| = \max\left\{\left|\frac{2\gamma\xi_k - \gamma^2\alpha_k - \alpha_k}{(1 + \gamma^2)\alpha_k}\right|, \left|\frac{2\gamma\xi_k - \gamma^2\alpha_k - \alpha_k}{\gamma c_k}\right| : k \in K\right\}.$$

By choosing $|c_k| \geq \gamma^{-1}(1 + \gamma^2)\alpha_k$, we find that $|||\,\tilde{Z}\,|||$ equals the asymptotic amplification factor ρ as given in Theorem 3.2. Hence, $|||\,\tilde{Z}^{\nu}\,||| \leq |||\,\tilde{Z}\,|||^{\nu} = \rho^{\nu}$. Obviously, we can never have strict inequality, so that we conclude that $|||\,\tilde{Z}^{\nu}\,||| = \rho^{\nu}$. Finally, it follows from (3.14) that $|||\,Z^{\nu}\,||| \leq \kappa(Q)|||\,\tilde{Z}^{\nu}\,||| = \kappa(Q)\rho^{\nu}$. Thus, the averaged amplification factor $\rho^{(\nu)}$ is bounded by $[\kappa(Q)]^{1/\nu}\rho$. This completes the proof of the theorem. ♦

We remark that for $\nu \to \infty$, we have the estimate $\rho^{(\nu)} \le \rho \max\{[\kappa(S(z))]^{1/\nu}: \mathrm{Re}(z) \le 0\}$, where $S(z)$ represents the eigensystem of $Z(z)$ and where we assumed that $Z(z)$ has distinct eigenvalues. The advantage of the estimate in Theorem 3.5 is that it holds for all ν.

3.3.2. The transformation matrix Q. In this subsection, we assume that the PILSRK method satisfies the conditions of Theorem 3.5. In order to obtain small amplification factors $(\rho^{(\nu)}, \sigma^{(\nu)})$ as defined by (3.13), we shall use the freedom left in choosing the transformation matrix Q. We specify our approach for the case where all eigenvalues $\xi_k \pm i\eta_k$ of A are complex ($\eta_k \ne 0$), so that $\sigma = s/2$. Then, the column vectors $\mathbf{q}_j$ of Q are defined by

$$(3.18)\quad \big(\mathbf{q}_{2k-1}, \mathbf{q}_{2k}\big) = \big(\beta_k\mathbf{x}_k + \delta_k\mathbf{y}_k,\ -\delta_k\mathbf{x}_k + \beta_k\mathbf{y}_k\big)\, Q_k, \quad k = 1, \dots, \tfrac{1}{2}\, s,$$

where β_k and δ_k are free parameters and $\mathbf{x}_k \pm i\mathbf{y}_k$ represent the normalized eigenvectors of A corresponding with $\xi_k \pm i\eta_k$ such that the first component of $\mathbf{y}_k$ vanishes. Here, Q_k is a transformation matrix satisfying (cf. (3.3))

$$(3.19\text{a})\quad \tilde{A}_{kk} = Q_k^{-1}\begin{pmatrix} \xi_k & \eta_k \\ -\eta_k & \xi_k \end{pmatrix} Q_k, \tilde{A}_{kk} := \begin{pmatrix} \gamma\alpha_k & \dfrac{\gamma(\gamma^2\alpha_k - 2\gamma\xi_k + \alpha_k)}{1 + \gamma^2} \\ -\dfrac{1 + \gamma^2}{\gamma}\alpha_k & 2\xi_k - \gamma\alpha_k \end{pmatrix}.$$

It can be verified that the matrix

$$(3.19\text{b})\quad Q_k = \frac{1}{\gamma(\gamma^2\alpha_k - 2\gamma\xi_k + \alpha_k)}\begin{pmatrix} (1 + \gamma^2)\eta_k & 0 \\ (1 + \gamma^2)(\gamma\alpha_k - \xi_k) & \gamma(\gamma^2\alpha_k - 2\gamma\xi_k + \alpha_k) \end{pmatrix}$$

satisfies (3.19a). By means of (3.18) and (3.19) it is easily verified that we do obtain the matrix $\tilde{A} = Q^{-1}AQ$. The advantage of this approach is that the resulting matrix Q has real entries.

For a given value of γ, the equations (3.19) and (3.20) determine a family of transformation matrices Q with free parameters vectors $\beta = (\beta_k)$, $\delta = (\delta_k)$ and $\mathbf{c} = (c_k)$, where $|c_k| \ge \gamma^{1}(1 + \gamma^2)\alpha_k$.

Table 3.2. Global amplification factors $\rho^{(\nu)}$ for PILSRK methods.

	4-stage Radau IIA corrector			8-stage Radau IIA corrector		
ν	PILSRK(D,I)	PILSRK(L,I)	PILSRK(T,Q)	PILSRK(D,I)	PILSRK(L,I)	PILSRK(T,Q)
1	3.60	0.59	1.95	19.83	1.03	3.51
2	2.48	0.54	0.98	11.52	0.94	1.88
3	1.64	0.53	0.76	7.74	0.91	1.30
4	1.16	0.53	0.66	5.55	0.90	1.19
5	0.96	0.52	0.61	4.08	0.89	1.09
…	…	…	…	…	…	…
9	0.72	0.51	0.53	1.86	0.88	0.87
10	0.69	0.51	0.52	1.72	0.88	0.85
11	0.67	0.51	0.51	1.61	0.88	0.83
…	…	…	…	…	…	…
∞	0.52	0.51	0.45	0.90	0.86	0.64

By a numerical search, we found in the case of the 4-stage and 8-stage Radau IIA correctors for $\gamma = \frac{7}{8}$ the values (3.20) yielding a sequence of satisfactory small amplification factors $\rho^{(\nu)}$ (see Table 3.2):

$$
(3.20) \quad \begin{aligned}
&s = 4: \beta = (5, -4),\ \ \delta = (-1, -5),\ \ \mathbf{a} = \tfrac{7}{8}\,(\alpha_1, \alpha_2)\,,\ \mathbf{c} = -\tfrac{113}{56}\,\mathbf{a},\\
&s = 8: \beta = (-0.9, -2, -2, 1.1),\quad \delta = (1.1, 0.3, 0.3, -1.9),\\
&\qquad\qquad \mathbf{a} = \tfrac{7}{8}\,(\alpha_1, \alpha_2, \alpha_3, \alpha_4)\,,\quad \mathbf{c} = -\tfrac{113}{56}\,\mathbf{a}.
\end{aligned}
$$

Table 3.2 also lists amplification factors $\rho^{(\nu)}$ for the PILSRK(L,I) and PILSRK(D,I) methods. This table clearly shows that in terms of $\rho^{(\nu)}$ values, the PILSRK(T,Q) methods are superior to the PILSRK(D,I) method. With respect to PILSRK(L,I), the $\rho^{(\nu)}$ values of PILSRK(T,Q) are smaller only for large numbers of inner iterations. In fact, they become less than those associated with PILSRK(L,I) only if ν is greater than about 10. However, in terms of the $\sigma^{(\nu)}$ values, the PILSRK(T,Q) methods are also superior to the PILSRK(L,I) method for *small* numbers of inner iterations, because in the case of PILSRK(T,Q), $\sigma^{(\nu)}$ becomes finite for $\nu > 2$, whereas PILSRK(L,I) has infinite $\sigma^{(\nu)}$-values for all ν.

4. The Newton-PILSRK iteration process

In actual application of the Newton-PILSRK iteration process {(2.1),(2.2)}, the inner iteration process will not always be iterated to convergence, so that the Newton iterates are only approximately computed. This will affect the convergence and stability behaviour and the computational costs of integration method.

4.1. Overall convergence and stability

The overall convergence of the Newton-PILSRK process is determined by the total number of inner iterations summed over all outer iterations in one step, that is, the effective amplification factors associated with the total iteration error $\mathbf{Y}^{(j,\nu)} - \mathbf{Y}_n$ are approximately given by $\rho^{(i)}$ and $\sigma^{(i)}$, where i denotes the total number of inner iterations needed to compute $\mathbf{Y}^{(j,\nu)}$, i.e. $i = (j-1)r + \nu$, and where r denotes the number of inner iterations per outer iteration. In order to see this, we define

$$(4.1)\ \ \mathbf{Y}^{(j,0)} := \mathbf{Y}^{(j-1,r)},\quad \mathbf{G}(\Delta) := \mathbf{F}\big(\mathbf{Y} + \Delta\big) - \mathbf{F}(\mathbf{Y}) - (I\otimes J)\Delta,\quad N := \big(I - A\otimes hJ\big)^{-1}(A\otimes I).$$

By a simple manipulation we find that

$$(4.2)\ \ \mathbf{Y}^{(j,\nu)} - \mathbf{Y}_n = M^{\nu}\Big(\mathbf{Y}^{(j-1,r)} - \mathbf{Y}_n\Big) + h(I - M^{r})N\mathbf{G}\big(\mathbf{Y}^{(j-1,r)} - \mathbf{Y}_n\big),\quad j = 1, \dots, m,$$

where M is defined in (3.1). Ignoring second-order terms, we may set $\mathbf{G}\big(\mathbf{Y}^{(j-1,r)} - \mathbf{Y}_n\big) = \mathbf{0}$, to obtain

$$(4.3)\ \ \mathbf{Y}^{(j,\nu)} - \mathbf{Y}_n = M^{i}\Big(\mathbf{Y}^{(0,r)} - \mathbf{Y}_n\Big),\quad i := (j-1)r + \nu\ .$$

From this relation, we see that in a first approximation, the convergence behaviour of the Newton-PILSRK iteration process is approximately characterized by the amplification factors. As a consequence, Table 3.2 applies if we replace ν by i.

A second feature of the overall performance of the integration method is its stability if the Newton iterates are *not exactly* computed. This aspect has been discussed in [12], where the number of iterations needed to achieve sufficient stability was computed. The values of mr for which the method becomes and remains L-stable depends on the predictor used. For the extrapolation (EPL) predictor defined by $\mathbf{Y}^{(0)} = (P\otimes I)\mathbf{Y}_{n-1}$, where P is such that $\mathbf{Y}^{(0)}$ has maximal order q = s-1, and the four-stage and eight-stage Radau IIA corrector, these stable mr-values are listed in Table 4.1. In the case of the four-stage corrector, the stable mr-values are acceptable for all three iteration strategies, but for the eight-stage corrector, only the Newton-PILSRK(T,Q) method possesses an acceptable stable mr-value.

Table 4.1. Stable values of mr for $\gamma = \frac{7}{8}$.

Iteration method	s = 4	s = 8
PILSRK(D,I)	7	> 61
PILSRK(L,I)	4	> 43
PILSRK(T,Q)	5	14

Summarizing, we conclude that with respect to the Newton-PILSRK(D,I)-based integration method, the Newton-PILSRK(T,Q) method always generates an integration method that has a superior convergence and stability behaviour. With respect to the Newton-PILSRK(L,I)-based integration method, we conclude that the Newton-PILSRK(T,Q) method:

(i) damps the stiff error components much stronger for i < s (Theorem 3.4),
(ii) has a better overall convergence for larger values of i (Table 3.2, with ν replaced by i),
(iii) is much more stable for the 8-stage corrector (Table 4.1).

4.2. Computational costs

In an actual implementation of the linear solver (2.2), we diagonalize (2.2) by a transformation $\mathbf{Y}^{(j,\nu)} = (S\otimes I)\mathbf{X}^{(j,\nu)}$ to obtain

$$(4.4)\quad \Big(I - S^{-1}BS\otimes hJ\Big)\Big(\mathbf{X}^{(j,\nu)} - \mathbf{X}^{(j,\nu-1)}\Big) = -\Big(I - S^{-1}AS\otimes hJ\Big)\mathbf{X}^{(j,\nu-1)} + (S^{-1}\otimes I)\mathbf{C}^{(j-1)},$$

where the matrix $S^{-1}BS$ is diagonal. For the PILSRK(L,I) and PILSRK(T($\gamma\neq1$),Q) methods, the matrices $S^{-1}BS$ and S corresponding to the 4-stage and 8-stage Radau IIA correctors are given in the Appendix to this paper. In this Appendix, we also give a computer-program type description of the Newton-PILSRK iteration process {(2.1),(2.2),(4.4)} and a specification of the computational costs of the most important steps of the algorithm. Here, we present in Table 4.2 the total costs per step for s-stage correctors where s is even. In this table, C_f and C_J denote the average costs of one component of $\mathbf{f}$ and its Jacobian J, respectively.

Table 4.2. Total computational costs per step.

Method	1 processor	$\frac{1}{2}$ s processors	s processors

PILSRK(L,I)
&
PILSRK(T(γ≠1),Q)
	$sd(\frac{2}{3}d^2 + \frac{d}{s}C_J + d + 2s)$
	$+ 4mrsd^2(1 + \frac{s}{2d})$
	$+ msd(s + C_f - 2d)$
		$2d(\frac{2}{3}d^2 + \frac{d}{s}C_J + d + 2s)$
		$+ 8mrd^2(1 + \frac{s}{2d})$
		$+ 2md(2s + C_f - 2d)$
			$d(\frac{2}{3}d^2 + \frac{d}{s}C_J + d + 2s)$
			$+ 4mrd^2(1 + \frac{s}{2d})$
			$+ md(s + C_f - 2d)$

PILSRK(T(1),Q)	$sd(\frac{1}{3}d^2 + \frac{d}{s}C_J + d + 2s)$ $+ 5mrsd^2(1 + \frac{2s}{5d})$ $+ msd(2s + C_f - 2d)$	$d(\frac{2}{3}d^2 + \frac{2d}{s}C_J + 2d + 4s)$ $+ 10mrd^2(1 + \frac{2s}{5d})$ $+ 2md(2s + C_f - 2d)$	$d(\frac{2}{3}d^2 + \frac{d}{s}C_J + d + 2s)$ $+ 8mrd^2(1 + \frac{s}{4d})$ $+ md(2s + C_f - 2d)$

The following conclusions can be drawn:

(i) Newton-PILSRK(L,I) and Newton-PILSRK(T(γ≠1),Q) are equally expensive,

(ii) If mr is fixed and $d > s + \frac{1}{2}C_f$, then the costs are minimized for r = 1,

(iii) Newton-PILSRK(L,I) and Newton-PILSRK(T(γ≠1),Q) are to be preferred on s processors, whereas Newton-PILSRK(T(γ=1),Q) is to be preferred on one or on σ processors.

5. Numerical illustration

In this section, we compare the new Newton-PILSRK(T(7/8),Q) method with the Newton-PILSRK(L,I) method. In our experiments, we use the EPL predictor defined in the preceding section and either the 4-stage or the 8-stage Radau IIA corrector with constant stepsizes. We integrated three test problems taken from the CWI test set [14]. In these problems, the initial condition was adapted such that the integration starts outside the transient phase. The first test problem is provided by a problem of Schäfer (called the HIRES problem in [7, p.157]). It consists of 8 mildly stiff nonlinear equations on the interval [5,305]. The second test example is the Pollution problem of Verwer [20]. The ODE system consists of 20 highly stiff nonlinear ODEs on the interval [5,60], originating from an air pollution model. Our third test problem, the Ring Modulator originating from circuit analysis, is a highly stiff system of 15 equations on the interval $[0,10^{-3}]$, and is due to Horneber [9].

The tables of results present the minimal number of correct digits cd of the components of **y** at the end point of the integration interval (i.e. at the end point, the absolute errors are written as 10^{-cd}). Negative cd-values are indicated with *. Table 5.1 leads us to the following conclusions:

(i) For fixed values of m ≥ 3, the Newton-PILSRK methods always converge and usually find the Newton iterate with high accuracy within two inner iterations (in the case of the 4-stage corrector, we even have convergence for m ≥ 1).

(ii) Comparing results for fixed values of mr reveals that r = 1 is usually preferable (however, in an actual implementation, m and r should both be determined dynamically, see also Remark 2.1).

(iii) For r ≤ 2 the Newton-PILSRK(T,Q) method is more robust than Newton-PILSRK(L,I), particularly for the eight-stage corrector, and approximates the Newton iterate usually much better (the better cd-values produced by Newton-PILSRK(L,I) in the Pollution problem for r = 2 and m = 3, 4 is due to 'overshoot' and does not mean that Newton-PILSRK(L,I) produces a better approximation to the corrector solution). The divergent behaviour is due to the development of instabilities for small values of mr (see Table 4.1).

Finally, we remark that for the relatively difficult Ring modulator problem, a parallel implementation of the Newton-PILSRK(L,I) method on the four-processor Cray-C98 / 4256 shows a speed-up ranging from at least 2.4 until at least 3.1 with respect to RADAU5 in one-processor mode (cf. [11]). Since Newton-PILSRK(T(γ≠1),Q) is equally expensive as Newton-PILSRK(L,I), the same speed-ups are expected for Newton-PILSRK(T(γ≠1),Q).

Table 5.1. Newton-PILSRK applied to HIRES with h = 15.

		4-stage Radau IIA corrector						8-stage Radau IIA corrector					
Solver	r	m=1	m=2	m=3	m=4	m=10	m=20	m=1	m=2	m=3	m=4	m=10	m=20
PILSRK(L,I)	1	*	3.0	4.8	5.1	7.3	7.9	*	*	*	*	8.2	9.9
PILSRK(T,Q)		*	4.5	4.9	5.3	7.7	7.9	*	*	*	*	9.3	10.8
PILSRK(L,I)	2	*	4.3	4.9	5.3	8.1	7.9	*	*	*	*	9.2	10.1
PILSRK(T,Q)		3.9	4.4	4.9	5.4	8.2	7.9	*	*	5.5	7.0	9.3	10.8
PILSRK(L,I)	10	3.8	4.4	4.9	5.4	8.2	7.9	*	*	5.6	7.0	9.4	10.3
PILSRK(T,Q)		3.8	4.4	4.9	5.4	8.2	7.9	*	*	5.6	7.0	9.3	10.8

Table 5.2. Newton-PILSRK applied to Pollution problem with h = 11.

		4-stage Radau IIA corrector						8-stage Radau IIA corrector					
Solver	r	m=1	m=2	m=3	m=4	m=10	m=20	m=1	m=2	m=3	m=4	m=10	m=20
PILSRK(L,I)	1	2.0	3.7	6.3	7.0	10.9	10.9	*	*	*	*	10.3	10.3
PILSRK(T,Q)		1.1	5.3	6.9	7.3	10.9	10.9	*	*	*	6.7	12.0	12.6
PILSRK(L,I)	2	4.6	5.7	7.5	8.5	10.9	10.9	*	*	*	8.0	10.3	10.7
PILSRK(T,Q)		4.9	5.7	6.7	7.9	10.9	10.9	*	2.9	6.6	7.8	12.6	12.3
PILSRK(L,I)	10	4.6	5.7	6.8	7.9	10.9	10.9	*	4.8	6.7	7.8	11.0	10.9
PILSRK(T,Q)		4.6	5.7	6.8	7.9	10.9	10.9	*	4.8	6.7	7.8	12.5	12.5

Table 5.3. Newton-PILSRK applied to the Ring Modulator with h = 1.25_{10}^{-7}.

		4-stage Radau IIA corrector						8-stage Radau IIA corrector					
Solver	r	m=1	m=2	m=3	m=4	m=10	m=20	m=1	m=2	m=3	m=4	m=10	m=20
PILSRK(L,I)	1	*	5.7	7.8	8.5	10.2	10.2	*	*	*	*	8.6	9.1
PILSRK(T,Q)		*	7.3	8.4	9.7	10.2	10.2	*	*	*	10.5	10.8	11.1
PILSRK(L,I)	2	5.5	7.5	8.7	10.2	10.2	10.2	*	*	*	8.9	9.4	9.3
PILSRK(T,Q)		5.7	7.4	8.8	10.0	10.2	10.2	*	8.5	10.4	10.9	10.5	11.3
PILSRK(L,I)	10	5.8	7.4	8.8	9.9	10.2	10.2	*	8.5	9.0	8.9	9.0	9.2
PILSRK(T,Q)		5.8	7.4	8.8	9.9	10.2	10.2	*	8.5	10.4	11.1	11.3	10.6

References

[1] Burrage, K. [1978]: A special family of Runge-Kutta methods for solving stiff differential equations, BIT 18, 22-41.

[2] Burrage, K. [1995]: Parallel and sequential methods for ordinary differential equations, Clarendon Press, Oxford.

[3] Burrage, K. & Chipman, F.H. [1989]: Construction of A-stable diagonally implicit multivalue methods, SIAM J. Numer. Anal. 26, 391-413.

[4] Burrage, K. & Suhartanto, H. [1997]: Parallel iterated methods based on multistep Runge-Kutta methods of Radau type, this issue, p. xx-yy.

[5] Butcher, J.C. [1976]: On the implementation of implicit Runge-Kutta methods, BIT 16, 237-240.

[6] Golub, G.H. & Van Loan, C.F. (1983): Matrix computations, North Oxford Academic, Oxford.

[7] Hairer, E. & Wanner, G. [1991]: Solving ordinary differential equations, II. Stiff and differential-algebraic problems, Springer-Verlag, Berlin.

[8] Hoffmann, W. & Swart, J.J.B. de [1995]: Approximating Runge-Kutta matrices by triangular matrices, Preprint NM-R9517, CWI, Amsterdam.

[9] Horneber, E.H. [1976]: Analysis of nonlinear RCLÜ-circuits by means of a mixed potential function with a systematic representation of the nonlinear dynamic circuit analysis (German), PhD thesis, University of Kaiserslautern.

[10] Houwen, P.J. van der, & Sommeijer, B.P. [1991]: Iterated Runge-Kutta methods on parallel computers, SIAM J. Sci. Stat. Comput. 12, 1000-1028.

[11] Houwen, P.J. van der & Swart, J.J.B. de [1996]: Triangularly implicit iteration methods for ODE-IVP solvers, to appear in SIAM. J. Sci. Comput.

[12] Houwen, P.J. van der, & Sommeijer, B.P. [1996]: CWI contributions to the development of parallel Runge-Kutta methods, to appear in APNUM.

[13] Lioen, W.M. [1996]: On the diagonal approximation of full matrices, to appear in J. Comput. Appl. Math.

[14] Lioen, W.M., Swart, J.J.B. de, & Veen, W.A. van der [1995]: Test set for IVP solvers, available via WWW at URL: http: //www.cwi.nl/cwi/projects/IVPtestset.shtml.

[15] Nevanlinna, O. [1985]: Matrix valued versions of a result of Von Neumann with an application to time discretization, J. Comput. Appl. Math. 12 & 13, 475-489.

[16] Nørsett, S.P. [1976]: Runge-Kutta methods with a multiple real eigenvalue only, BIT 16, 388-393.

[17] Orel, B. [1993]: Parallel Runge-Kutta methods with real eigenvalues, Appl. Numer. Math. 11, 241-250.

[18] Reichel, L. & Trefethen, L.N. (1992): Eigenvalues and pseudo-eigenvalues of Toeplitz matrices,Linear Algebra Appl. 162 / 164,153-185.

[19] Varga, R.S. [1962]: Matrix iterative analysis, Prentice Hall, Englewood Cliffs, N.J.

[20] Verwer, J.G. [1994]: Gauss-Seidel iteration for stiff ODEs from chemical kinetics, SIAM. J. Sci. Comput. 15, 1243-1250.

A. Appendix: Costs of PILSRK

In this appendix we specify the costs of the implementations of PILSRK(L,I) and PILSRK($T(\gamma)$,Q). In both methods the iterates satisfy the recursion

$$\begin{aligned}(I - S^{-1}BS \otimes hJ)(X^{(j,\nu)} - X^{(j,\nu-1)}) &= -(I - S^{-1}AS \otimes hJ)(X^{(j,\nu-1)} - X^{(j-1)}) \\ &\quad -X^{(j-1)} + h(S^{-1}A \otimes I)F(Y^{(j-1)}) + (E \otimes I)X_{n-1}.\end{aligned}$$

Here, $X_{n-1} = (S^{-1} \otimes I)Y_{n-1}$, $X^{(j,0)} = X^{(j-1)}$, $X^{(0)} = (S^{-1} \otimes I)P(Y_{n-1})$, $X^{(j)} = X^{(j,r)}$, $Y_n = (S \otimes I)X^{(m)}$, $P(\cdot)$ denotes the predictor operator, m the number of outer iterations and r the number of inner iterations. For PILSRK(L,I) and PILSRK($T(\gamma \neq 1)$,Q), the matrix $S^{-1}BS$ is diagonal, for PILSRK($T(1)$,Q), it is block diagonal, with 2×2 lower triangular blocks containing identical diagonal entries.

We implemented this recursion as:

```
        $Y_0 = (e \otimes I)y_0, \quad X_0 = (S^{-1} \otimes I)Y_0$
        for $n = 1, 2, \ldots, N$
(s1)        $LU = I - \mathrm{diag}(S^{-1}BS) \otimes hJ$
            $Y^{(0)} = P(Y_{n-1})$
(s2)        $X^{(0)} = (S^{-1} \otimes I)Y^{(0)}$
            for $j = 1, 2, \ldots, m$
(o1)            $R = X^{(j-1)} - h(S^{-1}A \otimes I)F(Y^{(j-1)}) - (E \otimes I)X_{n-1}$
(o2)            $X_i^{(j,1)} = X_i^{(j-1)} - (LU)_i^{-1}R_i$                                   (for $i$ odd)
(o3)            $X_i^{(j,1)} = X_i^{(j-1)} - (LU)_i^{-1}(R_i - b_{i,i-1}hJX_{i-1}^{(j,1)})$       (for $i$ even)
                for $\nu = 2, 3, \ldots, r$
(i1)                $H = (I - S^{-1}AS \otimes hJ)(X^{(j,\nu-1)} - X^{(j-1)}) - R$
(i2)                $X_i^{(j,\nu)} = X_i^{(j,\nu-1)} - (LU)_i^{-1}H_i$                           (for $i$ odd)
(i3)                $X_i^{(j,\nu)} = X_i^{(j,\nu-1)} - (LU)_i^{-1}(H_i - b_{i,i-1}hJX_{i-1}^{(j,\nu)})$   (for $i$ even)
                end
                $X^{(j)} = X^{(j,r)}$
(o4)            $Y^{(j)} = (S \otimes I)X^{(j)}$
            end
            $Y_n = Y^{(m)}, \quad X_n = X^{(m)}$
        end
```

Here, N is the number of integration steps. The Jacobian is assumed to be updated every time step. Notice that for PILSRK(L,I) and PILSRK($T(\gamma \neq 1)$,Q) the matrix $S^{-1}BS$ is diagonal, so that one can omit (**o3**) and (**i3**) for this case, if one performs (**o2**) and (**i2**) for all i. For PILSRK($T(1)$,Q) we only need σ processors to perform the LU-decompositions in parallel, where σ is the number of complex conjugated eigenvalue pairs. Here we assume that s is even, so $\sigma = s/2$.

The following tables list the costs of the most important steps of this algorithm. As before, d is the dimension of the problem. The average costs of one component of the right-hand-side function $\mathbf{f}$ and one entry of its Jacobian J are denoted by $C_{\mathbf{f}}$ and C_J, respectively. The Jacobian is assumed to be full. In the first column the computation that has to be performed is listed. The second column gives the number of floating point operations required for this computation if only one processor is available. The sequential costs of the compuation on σ and s processors can be found in the third and fourth column, respectively. For reasons of simplicity, we did not exploit the lower triangul form of the matrix S in PILSRK(L,I), nor the block diagonal form of the matrix $S^{-1}AS$ in PILSRK($T(\gamma)$,Q).

PILSRK(L,I) & PILSRK($T(\gamma \neq 1)$,Q):

Computation	**Costs (flops)**		
	on 1 processor	on σ processors	on s processors
(**s1**)	$sd^2(\frac{2}{3}d + \frac{1}{s}\, C_J + 1)$	$2d^2(\frac{2}{3}d + \frac{1}{s}\, C_J + 1)$	$d^2(\frac{2}{3}d + \frac{1}{s}\, C_J + 1)$
(**s2**)	$2s^2d$	$4sd$	$2sd$
(**o1**)	$sd(2s + C_{\mathbf{f}})$	$2d(2s + C_{\mathbf{f}})$	$d(2s + C_{\mathbf{f}})$
(**o2**) $(\forall s)$	$2sd^2$	$4d^2$	$2d^2$
(**i1**)	$2sd(d + s)$	$4d(d + s)$	$2d(d + s)$
(**i2**) $(\forall s)$	$2sd^2$	$4d^2$	$2d^2$
(**o4**)	s^2d	$4sd$	$2sd$
Total per time step	$sd(\frac{2}{3}d^2 + \frac{d}{s}\, C_J + d + 2s)$ $+sdm(2d + 3s + C_{\mathbf{f}}$ $+(r-1)(4d + 2s))$	$2d(\frac{2}{3}d^2 + \frac{d}{s}\, C_J + d + 2s)$ $+2dm(2d + 4s + C_{\mathbf{f}}$ $+(r-1)(4d + 2s))$	$d(\frac{2}{3}d^2 + \frac{d}{s}\, C_J + d + 2s)$ $+dm(2d + 4s + C_{\mathbf{f}}$ $+(r-1)(4d + 2s))$

PILSRK($T(1)$,Q):

Computation	**Costs (flops)**		
	on 1 processor	on σ processors	on s processors
(**s1**)	$sd^2(\frac{1}{3}d + \frac{1}{s}\, C_J + 1)$	$2d^2(\frac{1}{3}d + \frac{1}{s}\, C_J + 1)$	$d^2(\frac{2}{3}d + \frac{1}{s}\, C_J + 1)$
(**s2**)	$2s^2d$	$4sd$	$2sd$
(**o1**)	$sd(2s + C_{\mathbf{f}})$	$2d(2s + C_{\mathbf{f}})$	$d(2s + C_{\mathbf{f}})$
(**o2**)	sd^2	$2d^2$	$2d^2$
(**o3**)	$2sd^2$	$4d^2$	$4d^2$
(**i1**)	$2sd(d + s)$	$4d(d + s)$	$2d(d + s)$
(**i2**)	sd^2	$2d^2$	$2d^2$
(**i3**)	$2sd^2$	$4d^2$	$4d^2$
(**o4**)	$2s^2d$	$4sd$	$2sd$
Total per time step	$sd(\frac{1}{3}d^2 + \frac{d}{s}\, C_J + d + 2s)$ $+sdm(3d + 4s + C_{\mathbf{f}}$ $+(r-1)(5d + 2s))$	$2d(\frac{1}{3}d^2 + \frac{d}{s}\, C_J + d + 2s)$ $+2dm(3d + 4s + C_{\mathbf{f}}$ $+(r-1)(5d + 2s))$	$d(\frac{2}{3}d^2 + \frac{d}{s}\, C_J + d + 2s)$ $+dm(6d + 4s + C_{\mathbf{f}}$ $+(r-1)(8d + 2s))$

B. Method parameters

In this appendix we specify the method parameters of the PILSRK(L,I) and PILSRK($T(7/8)$,Q) methods for $s = 4$ and $s = 8$. We list the matrices $S^{-1}BS$ and S, which are needed for the implementation of formula (4.4). As additional information we provide B, the matrix that approximates A.

PILSRK(L,I):

$$\boxed{s = 4}$$

$$\mathrm{diag}(S^{-1}BS) \;=\; \begin{pmatrix} 0.1130 & 0.2905 & 0.3083 & 0.1176 \end{pmatrix}$$

$$S \;=\; \begin{pmatrix} 1.0000 & 0 & 0 & 0 \\ -1.3205 & 1.0000 & 0 & 0 \\ 2.1594 & -27.2263 & 1.0000 & 0 \\ -119.8988 & -66.8265 & 2.3158 & 1.0000 \end{pmatrix}$$

$$B \;=\; \begin{pmatrix} 0.1130 & 0 & 0 & 0 \\ 0.2344 & 0.2905 & 0 & 0 \\ 0.2167 & 0.4834 & 0.3083 & 0 \\ 0.2205 & 0.4668 & 0.4414 & 0.1176 \end{pmatrix}$$

$$\boxed{s = 8}$$

$$\mathrm{diag}(S^{-1}BS) \;=\; \begin{pmatrix} 0.0288 & 0.0865 & 0.1345 & 0.1624 & 0.1654 & 0.1427 & 0.0976 & 0.0308 \end{pmatrix}$$

$$S \;=\; \begin{pmatrix} 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.0694 & 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.0486 & -3.2354 & 1.0000 & 0 & 0 & 0 & 0 & 0 \\ -1.0718 & 7.7636 & -8.1101 & 1.0000 & 0 & 0 & 0 & 0 \\ 1.1852 & -19.0240 & 62.0182 & -88.1175 & 1.0000 & 0 & 0 & 0 \\ -1.4887 & 62.7656 & -0.1720e4 & -0.1141e4 & 11.3694 & 1.0000 & 0 & 0 \\ 2.4708 & -908.4889 & -0.9526e4 & -0.4070e4 & 39.2573 & 4.7028 & 1.0000 & 0 \\ -88.2154 & -2.0073e3 & -1.5590e4 & -0.6097e4 & 58.3751 & 7.4699 & 2.0027 & 1.0000 \end{pmatrix}$$

$$B \;=\; \begin{pmatrix} 0.0288 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0617 & 0.0865 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0553 & 0.1553 & 0.1345 & 0 & 0 & 0 & 0 & 0 \\ 0.0583 & 0.1424 & 0.2261 & 0.1624 & 0 & 0 & 0 & 0 \\ 0.0567 & 0.1483 & 0.2106 & 0.2619 & 0.1654 & 0 & 0 & 0 \\ 0.0575 & 0.1454 & 0.2171 & 0.2471 & 0.2572 & 0.1427 & 0 & 0 \\ 0.0571 & 0.1467 & 0.2144 & 0.2522 & 0.2460 & 0.2124 & 0.0976 & 0 \\ 0.0573 & 0.1463 & 0.2151 & 0.2510 & 0.2483 & 0.2073 & 0.1338 & 0.0308 \end{pmatrix}$$

PILSRK($T(7/8)$,Q):

$$\boxed{s = 4}$$

$$\mathrm{diag}(S^{-1}BS) = \begin{pmatrix} 0.1521 & 0.1986 & 0.1737 & 0.2269 \end{pmatrix}$$

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

$$B = \begin{pmatrix} 0.1096 & -0.0430 & 0.0268 & -0.0080 \\ 0.2085 & 0.3064 & -0.0671 & 0.0211 \\ 0.2484 & 0.0823 & 0.2573 & -0.0142 \\ 0.2596 & -0.0515 & 0.4219 & 0.0780 \end{pmatrix}$$

$$\boxed{s = 8}$$

$$\mathrm{diag}(S^{-1}BS) = \begin{pmatrix} 0.0679 & 0.0886 & 0.0768 & 0.1003 & 0.0823 & 0.1074 & 0.0849 & 0.1109 \end{pmatrix}$$

$$S = \begin{pmatrix} 0.1430 & 0.0149 & 0.0051 & -0.0013 & -0.0208 & -0.0029 & 0.0180 & -0.0001 \\ -0.2667 & -0.0284 & -0.0306 & -0.0006 & 0.0195 & 0.0034 & -0.0182 & -0.0002 \\ 0.4848 & 0.0540 & 0.0915 & 0.0083 & -0.0050 & -0.0010 & 0.0205 & -0.0008 \\ -0.8881 & -0.1065 & -0.0372 & -0.0099 & 0.0975 & 0.0101 & -0.0112 & -0.0072 \\ 1.1326 & 0.1628 & -0.9048 & -0.0996 & 0.2347 & -0.0050 & -0.1102 & -0.0522 \\ 1.6603 & 0.1105 & -0.2681 & 0.0933 & -1.0125 & -0.2481 & -1.3834 & -0.2826 \\ -5.9025 & -0.7539 & 7.6108 & 1.0254 & -7.3467 & -1.0128 & -6.3367 & -0.8981 \\ -8.9828 & -0.9978 & 14.1609 & 1.6360 & -14.1886 & -1.6730 & -12.2810 & -1.4897 \end{pmatrix}$$

$$B = \begin{pmatrix} 0.0507 & -0.0264 & -0.0147 & -0.0077 & 0.0061 & -0.0034 & 0.0022 & -0.0008 \\ 0.0295 & 0.0856 & 0.0153 & 0.0162 & -0.0104 & 0.0059 & -0.0037 & 0.0014 \\ 0.0513 & 0.1372 & 0.0952 & -0.0314 & 0.0170 & -0.0096 & 0.0059 & -0.0022 \\ 0.1601 & 0.0455 & 0.0662 & 0.1458 & -0.0342 & 0.0201 & -0.0127 & 0.0048 \\ 0.2072 & 0.0253 & 0.0569 & 0.0462 & 0.1460 & -0.0312 & 0.0131 & -0.0034 \\ 0.2495 & -0.0151 & 0.0590 & 0.0185 & 0.1461 & 0.0202 & 0.0634 & -0.0262 \\ 0.2568 & -0.0281 & 0.0923 & -0.0159 & 0.0405 & 0.0418 & 0.2095 & -0.0688 \\ 0.2653 & -0.0325 & 0.0873 & -0.0924 & 0.1092 & 0.0499 & 0.2190 & -0.0340 \end{pmatrix}$$