Numerical Algorithms 18 (1998) 233–257

Splitting methods for second-order initial value problems

P.J. van der Houwen [a] and E. Messina [b]

[a] *CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands*
[b] *Dipartimento di Matematica e Applicazioni R. Caccioppoli, University of Naples Federico II, Via Cintia, I-80126 Naples, Italy*

Received 27 November 1997; revised 17 July 1998
Communicated by C. Brezinski

We consider implicit integration methods for the solution of stiff initial value problems for second-order differential equations of the special form $\mathbf{y}'' = \mathbf{f}(\mathbf{y})$. In implicit methods, we are faced with the problem of solving systems of implicit relations. This paper focuses on the construction and analysis of iterative solution methods which are effective in cases where the Jacobian of the right-hand side of the differential equation can be split into a sum of matrices with a simple structure. These iterative methods consist of the modified Newton method and an iterative linear solver to deal with the linear Newton systems. The linear solver is based on the approximate factorization of the system matrix associated with the linear Newton systems. A number of convergence results are derived for the linear solver in the case where the Jacobian matrix can be split into commuting matrices. Such problems often arise in the spatial discretization of time-dependent partial differential equations. Furthermore, the stability matrix and the order of accuracy of the integration process are derived in the case of a *finite* number of iterations.

Keywords: second-order partial differential equations, splitting methods, approximate factorization

1. Introduction

We consider initial value problems (IVPs) for systems of second-order ordinary differential equations (ODEs) of the special form

$$\frac{\mathrm{d}^2\mathbf{y}(t)}{\mathrm{d}t^2} = \mathbf{f}\big(\mathbf{y}(t)\big), \quad \mathbf{y}, \mathbf{f} \in \mathbb{R}^d. \tag{1.1}$$

We shall assume that the IVP for equation (1.1) is *stiff*. In analogy with the definition of stiff IVPs for *first-order* ODEs (see, e.g., [6, p. 2]), we shall call IVPs for equation (1.1) stiff if "explicit integration methods do not work". In order to make this more precise, we should indicate when explicit methods do not work. The success of explicit methods largely depends on the stepsize h used and the spectral radius ρ of the Jacobian matrix $\partial\mathbf{f}/\partial\mathbf{y}$ of the right-hand side function. In fact, due to their relatively small stability region, explicit methods can only work if the stepsize is such that in the neighbourhood of the exact solution the values of $h^2\rho(\partial\mathbf{f}/\mathrm{d}\mathbf{y})$ are sufficiently small. This leads us to the

following more quantitative definition of stiffness. Let h be such that the exact solution of (1.1) can be represented with sufficient accuracy on the discrete grid $\{t_n = t_{n-1}+h\}$. Then (1.1) is called stiff if in the neighbourhood of the exact solution $h^2\rho(\partial\mathbf{f}/\mathrm{d}\mathbf{y}) \gg 1$. As a consequence, applying explicit methods to stiff IVPs means that we need stepsizes that are determined by stability conditions rather than by accuracy considerations (we have a similar situation in the case of stiff first-order ODEs). Thus, if the IVP for (1.1) is stiff, then the method to be used should preferably be implicit.

Stiff equations of the form (1.1) often arise if time-dependent partial differential equations (PDEs) are semidiscretized by the method of lines (examples will be given in section 2.1). Solving such equations by an implicit method implies that we are faced with the problem of solving systems of implicit relations. This paper focuses on the construction and analysis of iterative solution methods which are effective in cases where an approximation J to $\partial\mathbf{f}/\partial\mathbf{y}$ can be split into a sum of σ-matrices J_i such that the matrices J_i have an essentially simpler structure than the matrix J (in sections 2.1 and 3.2, we will indicate what is meant by an "essentially simpler structure"). These iterative methods consist of the modified Newton method (the outer iteration), in which the linear Newton systems are solved by a second iteration process (the inner iteration) which is based on approximate factorization. The inner–outer iteration process is called *approximate factorization iteration* or briefly AF iteration.

In [7] AF iteration was used for solving fully implicit discretizations of transport models and in [3] AF iteration was analyzed in the case of a large class of implicit integration methods for systems of first-order ODEs originating from the semidiscretization of PDEs. In the latter paper, general convergence and stability results are presented. These results can also be used for second-order ODE methods by writing (1.1) as a first-order system and by simply integrating this system by a first-order ODE solver (the black box approach). Unfortunately, in the usual case where the eigenvalues of $\partial\mathbf{f}/\partial\mathbf{y}$ are negative, the convergence and stability properties of the black box approach are quite poor, because the special structure of the first-order form of (1.1) is not exploited. To illustrate this, consider a Runge–Kutta (RK) method for first-order ODEs $\mathbf{y}' = \mathbf{g}(\mathbf{y})$, let the Butcher matrix $\tilde{A}$ of the RK method be a matrix with *complex* eigenvalues, and suppose that $\partial\mathbf{g}/\partial\mathbf{y}$ can be written as the sum of two commuting matrices K_1 and K_2. Then it can be shown that the approximate factorization iteration process cannot be unconditionally convergent if the eigenvalues of K_1 and K_2 are purely imaginary (see [3]). Now we apply the same RK method to the first-order form of (1.1). Suppose that the Jacobian associated with the right-hand side of (1.1) can be split into two matrices J_1 and J_2 which share the same eigensystem with negative eigenvalues (for example, this happens if (1.1) originates from the spatial discretization of a two-dimensional wave equation, see section 2.1). Then, the matrices K_1 and K_2 associated with the first-order form $\mathbf{y}' = \mathbf{g}(\mathbf{y})$ of (1.1) commute and their eigenvalues are purely imaginary. Hence, as we mentioned above, the AF iteration process for solving the implicit RK relations will not be unconditionally convergent. However, exploiting the special structure of the first-order form $\mathbf{y}' = \mathbf{g}(\mathbf{y})$ of (1.1), the implicit RK relations can be simplified (see section 2 for details) and applying AF iteration

to these simplified relations, we obtain unconditional convergence provided that the eigenvalues $\lambda(\widetilde{A})$ of the underlying Butcher matrix $\widetilde{A}$ satisfy $|\arg(\lambda(\widetilde{A}))| \leqslant \pi/4$. Examples are the Butcher matrices of the third-order Radau IIA, the fourth-order Lobatto IIIA, and the fourth-order and sixth-order Gauss methods. Thus, although the *solutions* of the original and the simplified RK relations are identical, the *convergence properties* of AF iteration are quite different.

The purpose of this paper is

(i) to see to what extent the convergence and stability results valid for first-order ODE methods change in the second-order case (1.1), and

(ii) to select from a wide class of second-order ODE methods suitable methods for solving stiff IVPs.

The second-order ODE methods considered here belong to the class of so-called General Linear Methods (GLMs). For first-order ODEs, such methods have been introduced by Butcher in 1966 (see [1, p. 335] for a detailed discussion). In section 2, we show that GLM methods can be defined in a similar way for second-order ODEs given by (1.1). The advantage of using the GLM format is that almost any IVP solver can be written as a GLM, so that the analysis developed in this paper applies to a wide variety of methods. Section 3 discusses the structure of the implicit relations arising in these GLMs and defines the outer–inner iteration process for the implicit stage values. In section 4, a number of convergence results are derived for the model situation where the matrices J_i share the same eigensystem and possess a negative eigenvalue spectrum. Finally, section 5 presents order of accuracy and stability results in the case of a *finite* number of inner and outer iterations.

2. Preliminaries

In this section, we present examples of stiff ODEs of the form (1.1) originating from time-dependent PDEs and examples of implicit second-order ODE methods using the GLM formularium.

2.1. *Examples of stiff second-order ODEs*

Our first example is the so-called *equation of telegraphy* (cf., e.g., [2, p. 15])

$$\frac{\partial^2 u}{\partial t^2} = c^2 \Delta_\mu u + \frac{1}{4} k^2 u + g(t, x_1, \ldots, x_\mu), \qquad \Delta_\mu := \frac{\partial^2}{\partial x_1^2} + \cdots + \frac{\partial^2}{\partial x_\mu^2},$$
$$0 \leqslant t \leqslant 1, \ 0 \leqslant x_i \leqslant 1 \tag{2.1}$$

with initial conditions at $t = 0$ and boundary conditions along the spatial boundaries. Here, k is the friction constant, μ denotes the number of spatial dimensions, g is a given forcing function, and $u = \phi \exp(kt/2)$, where ϕ denotes a disturbance that is propagated with velocity c in the $(x_1, \ldots, x_\mu)$-plane. For example, this equation

describes the displacement of a string or membrane, electro-magnetic waves, and long low water waves (shallow water waves).

Replacing the spatial domain by a finite set of grid points and applying the method of lines yields a system of ODEs of the form

$$\frac{\mathrm{d}^2\mathbf{y}(t)}{\mathrm{d}t^2} = c^2\left(X_1 + \cdots + X_\mu + \frac{k^2}{4c^2}\right)\mathbf{y}(t) + \mathbf{g}(t), \tag{2.2}$$

where $\mathbf{y}$ approximates u at the grid points and $\mathbf{g}$ is a vector taking the inhomogeneous part of (2.1) and the boundary conditions into account. Let us consider the case of Dirichlet boundary conditions. Then, all rows of the matrix X_i represent discretizations of the differential operator $\partial^2/\partial x_i^2$. If the method of lines is based on standard, second-order, symmetric differences with spatial mesh Δx, then X_i can be characterized by the stencil $(\Delta x)^{-2}$ $[1, -2, 1]$. Such matrices possess eigenvalues in the interval $[-4(\Delta x)^{-2}, -\pi^2]$, so that $\rho(\partial\mathbf{f}/\mathrm{d}\mathbf{y}) = \mathrm{O}((\Delta x)^{-2})$. Hence, if the exact solution of the IVP for (2.1) is such that a discrete representation of this solution requires a space–time grid in which the spatial mesh Δx is much smaller than the time steps h, then $h^2\rho(\partial\mathbf{f}/\mathrm{d}\mathbf{y}) = \mathrm{O}(h^2(\Delta x)^{-2}) \gg 1$, i.e., the IVP is stiff. For example, this happens if (2.1) models long shallow water waves (see, e.g., [11, p. 142]). Furthermore, the Jacobian of (2.2) is given by $\partial\mathbf{f}/\partial\mathbf{y} = c^2(X_1 + \cdots + X_\mu) + (k/2)^2$ which can be split into a sum of μ matrices $J_i = c^2X_i + \mu^{-1}(k/2)^2 I$, where I denotes the identity matrix. These matrices J_i are all characterized by one-dimensional 3-point stencils, whereas $\partial\mathbf{f}/\partial\mathbf{y}$ is characterized by a μ-dimensional $(2\mu+1)$-point stencil. This feature can be exploited in the AF iteration process (see section 3.2).

Our second example is the equation for the transverse motion of a bar or plate given by [12, p. 54]

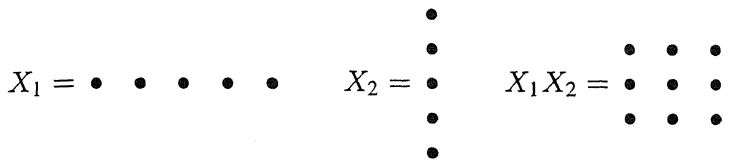
$$\frac{\partial^2 u}{\partial t^2} = -c^2(\Delta_\mu)^2 u + g(t, x_1, \ldots, x_\mu), \quad 0 \leqslant t \leqslant 1,\ 0 \leqslant x_i \leqslant 1, \tag{2.3}$$

where Δ_μ is again the Laplacian as defined in (2.1), c some constant, and g the forcing function. Proceeding as described above, we obtain the ODE system

$$\frac{\mathrm{d}^2\mathbf{y}(t)}{\mathrm{d}t^2} = -c^2(X_1 + \cdots + X_\mu)^2\mathbf{y}(t) + \mathbf{g}(t), \tag{2.4}$$

where the X_i are the same matrices as in (2.2). Here, $\rho(\partial\mathbf{f}/\mathrm{d}\mathbf{y}) = \mathrm{O}((\Delta x)^{-4})$. Hence, if the exact solution of the IVP for (2.3) is such that its discrete representation requires a space–time grid in which the spatial mesh Δx is of about the same size as the time steps h, then $h^2\rho(\partial\mathbf{f}/\mathrm{d}\mathbf{y}) = \mathrm{O}(h^2(\Delta x)^{-4}) = \mathrm{O}((\Delta x)^{-2})$. Evidently, this implies that the stiffness of the IVP increases as Δx decreases. In order to see the properties of the splitting of the Jacobian $\partial\mathbf{f}/\partial\mathbf{y} = -c^2(X_1 + \cdots + X_\mu)^2$, we first consider the 2-di-

mensional case, where $\partial \mathbf{f}/\partial \mathbf{y} = -c^2(X_1^2 + 2X_1X_2 + X_2^2)$. The stencils characterizing the matrices X_1, X_2 and X_1X_2 have the generic form



Hence, the Jacobian $\partial \mathbf{f}/\partial \mathbf{y}$ is characterized by a 13-point stencil. Setting $J_1 = -c^2X_1^2$, $J_2 = -c^2X_2^2$ and $J_3 = -2c^2X_1X_2$, we see that the Jacobian can be split into less complicated matrices, viz. two 5-point stencils and one 9-point stencil. Similarly, in the 3-dimensional case the Jacobian is given by a 33-point stencil which can be split into three 5-point stencils and three 9-point stencils. Again, these features can be exploited when applying the AF iteration process (see section 3.2).

2.2. *General linear methods*

A direct extension of the GLMs of Butcher to equations of the second-order form (1.1) reads

$$\mathbf{U}_{n+1} = (R \otimes I)\mathbf{U}_n + h^2(S \otimes I)\mathbf{F}(\mathbf{U}_n) + h^2(T \otimes I)\mathbf{F}(\mathbf{U}_{n+1}), \quad n = 1, 2, \ldots . \tag{2.5}$$

Here R, S and T denote $k \times k$ matrices, I is the $d \times d$ identity matrix, h is the stepsize $t_{n+1} - t_n$, and $\otimes$ denotes the Kronecker product, i.e., if $R = (r_{ij})$, then $R \otimes I$ denotes a block matrix with blocks $r_{ij}I$. In this paper, we assume that each of the k components $\mathbf{u}_{n+1,i}$ of the kd-dimensional solution vector $\mathbf{U}_{n+1}$ represents a numerical approximation either to the exact solution vector $\mathbf{y}(t_n + a_ih)$ or to the exact derivative vector $h\mathbf{y}'(t_n + a_ih)$. The vector $\mathbf{a} := (a_i)$ is called the *abscissa vector*, the quantities $\mathbf{U}_{n+1}$ the *stage vectors* and their components $\mathbf{u}_{n+1,i}$ the *stage values*. The stage values approximating $\mathbf{y}(t_n + a_ih)$ will be called *solution values* and those approximating $h\mathbf{y}'(t_n + a_ih)$ *derivative values*. Furthermore, for any vector $\mathbf{U}_n = (\mathbf{u}_{ni})$, $\mathbf{F}(\mathbf{U}_n)$ contains the right-hand side values $(\mathbf{f}(\mathbf{u}_{ni}))$.

The GLM (2.5) is completely determined by the arrays $\{\mathbf{a}, R, S, T\}$. Given the starting vector $\mathbf{U}_1$, (2.5) defines a sequence of vectors $\mathbf{U}_2$, $\mathbf{U}_3$, $\mathbf{U}_4, \ldots$, from which approximations to the exact solution values can be obtained.

It may happen that R and S have zero columns for the same column index j. In such cases, the jth component $\mathbf{u}_{1,j}$ of $\mathbf{U}_1$ is not needed to start the integration process. All stage values that we do need to start the method are called *external* stage values, otherwise they are called *internal* stage values (cf. Butcher [1, p. 367]). The distinction between internal and external stage values is needed in the stability analysis given in section 5.

In this paper, we shall assume that one or more abscissae a_i equal 1. If the corresponding components $\mathbf{u}_{n+1,i}$ of $\mathbf{U}_{n+1}$ are external stage values, then these components will be called *step point values* (the points $\{t_0, t_1, \ldots\}$ are called *step points*). A stage

value $\mathbf{u}_{n+1,i}$ which provides an approximation to the exact solution value $\mathbf{y}(t_n + a_i h)$ is said to be accurate of order p if for sufficiently smooth right-hand side functions $\mathbf{f}$ and for all points $\{t_n + a_i h,\ n = 0, 1, \ldots\}$, we have that $\mathbf{u}_{n+1,i} = \mathbf{y}(t_n + a_i h) + \mathrm{O}(h^p)$. The *maximal* order of accuracy of the step point values is called the *step point order*.

Of course, the second-order ODE (1.1) can also be solved by reducing the ODE (1.1) to first-order form and by application of a first-order ODE method. There are now two options:

(i) the *black box* approach where the first-order ODE method is used as a black box method, or

(ii) the *indirect* second-order ODE-method approach where the first-order ODE method is rewritten as a second-order ODE method by exploiting the special structure of the first-order ODE system.

In the black box option, we have to rely on the properties of the first-order ODE method, including the properties of the iteration process implemented for solving the implicit relations. Since it is often more advantageous, with respect to numerical performance, to follow the indirect second-order ODE-method option, we explicitly derive the resulting second-order ODE method. Let us write (1.1) as $\mathbf{y}' = \mathbf{z}$, $\mathbf{z}' = \mathbf{f}(\mathbf{y})$ and let us apply a GLM defined by the arrays $(\tilde{\mathbf{a}}, \widetilde{R}, \widetilde{S}, \widetilde{T})$. It can be verified that the resulting method is equivalent with separately applying this GLM to $\mathbf{y}' = \mathbf{z}$ and to $\mathbf{z}' = \mathbf{f}(\mathbf{y})$. Hence, let us associate with $\mathbf{y}$ and $\mathbf{z}$ the stage vectors $\mathbf{Y}$ and $\mathbf{Z}$. Then, $\mathbf{Y}$ and $\mathbf{Z}$ satisfy

$$\mathbf{Y}_{n+1} = (\widetilde{R} \otimes I)\mathbf{Y}_n + h(\widetilde{S} \otimes I)\mathbf{Z}_n + h(\widetilde{T} \otimes I)\mathbf{Z}_{n+1},$$
$$\mathbf{Z}_{n+1} = (\widetilde{R} \otimes I)\mathbf{Z}_n + h(\widetilde{S} \otimes I)\mathbf{F}(\mathbf{Y}_n) + h(\widetilde{T} \otimes I)\mathbf{F}(\mathbf{Y}_{n+1}).$$

By substitution of the second equation into the first and by defining the extended stage vector $\mathbf{U}_n := (\mathbf{Y}_n^{\mathrm{T}}, h\mathbf{Z}_n^{\mathrm{T}})^{\mathrm{T}}$, we obtain a GLM for second-order ODEs (see also Hairer [5])

$$\mathbf{a} = \begin{pmatrix} \tilde{\mathbf{a}} \\ \tilde{\mathbf{a}} \end{pmatrix}, \qquad R = \begin{pmatrix} \widetilde{R} & \widetilde{S} + \widetilde{T}\widetilde{R} \\ \mathrm{O} & \widetilde{R} \end{pmatrix}, \qquad S = \begin{pmatrix} \widetilde{T}\widetilde{S} & \mathrm{O} \\ \widetilde{S} & \mathrm{O} \end{pmatrix},$$
$$T = \begin{pmatrix} \widetilde{T}^2 & \mathrm{O} \\ \widetilde{T} & \mathrm{O} \end{pmatrix}. \tag{2.6}$$

Note that in (2.6) only $\mathbf{Y}_{n+1}$ is implicitly defined and should be solved by some iteration process. Thus, this iteration process needs to be applied to only kd implicit relations. This is a direct consequence of the special structure of the first-order system. Ignoring this special structure, that is, applying the black box option (i), would lead to iteration of $2kd$ implicit relations. Of course, if the iteration processes used in the two options both converge, then they converge to the same numerical solution. However, it will turn out that the iteration process in the indirect second-order ODE-method approach often converges where it does not converge in the black box approach.

Example 2.1. An example of a GLM of the form (2.6) with step point order $p = 2$ is the GLM:

$$\mathbf{a} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \qquad R = \frac{1}{9}\begin{pmatrix} 0 & 9 & 0 & 0 \\ -3 & 12 & -2 & 8 \\ 0 & 0 & 0 & 9 \\ 0 & 0 & -3 & 12 \end{pmatrix}, \qquad S = \mathrm{O},$$
$$T = \frac{1}{9}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \end{pmatrix} \tag{2.7}$$

derived from the two-step backward differentiation method (BDM). Here, $\mathbf{U}_{n+1}$ approximates

$$\big(\mathbf{y}(t_n)^{\mathrm{T}}, \mathbf{y}(t_n + h)^{\mathrm{T}}, h\mathbf{y}'(t_n)^{\mathrm{T}}, h\mathbf{y}'(t_n + h)^{\mathrm{T}}\big)^{\mathrm{T}}.$$

Example 2.2. Another *indirect* second-order ODE method, derived from the 2-stage Radau IIA-based method for first-order ODEs, is defined by the Runge–Kutta–Nyström (RKN) method:

$$\mathbf{a} = \begin{pmatrix} 1/3 \\ 1 \\ 1 \end{pmatrix}, \qquad R = \frac{1}{3}\begin{pmatrix} 0 & 3 & 1 \\ 0 & 3 & 3 \\ 0 & 0 & 3 \end{pmatrix}, \qquad S = \mathrm{O},$$
$$T = \frac{1}{36}\begin{pmatrix} 4 & -2 & 0 \\ 18 & 0 & 0 \\ 27 & 9 & 0 \end{pmatrix}, \tag{2.8}$$

where

$$\mathbf{U}_{n+1} \approx \big(\mathbf{y}(t_n + h/3)^{\mathrm{T}}, \mathbf{y}(t_n + h)^{\mathrm{T}}, h\mathbf{y}'(t_n + h)^{\mathrm{T}}\big)^{\mathrm{T}}.$$

This method has step point order 3.

Example 2.3. A *direct* second-order ODE method is given by (cf. Sharp et al. [10]):

$$\mathbf{a} = \begin{pmatrix} 17/14 \\ 23/60 \\ 1 \\ 1 \end{pmatrix}, \qquad R = \begin{pmatrix} 0 & 0 & 1 & 17/14 \\ 0 & 0 & 1 & 23/60 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad S = \mathrm{O},$$
$$T = \begin{pmatrix} 289/392 & 0 & 0 & 0 \\ -234179/352800 & 289/392 & 0 & 0 \\ -21/698 & 185/349 & 0 & 0 \\ 49/349 & 300/349 & 0 & 0 \end{pmatrix}, \tag{2.9}$$

where $\mathbf{U}_{n+1}$ approximates

$$\left(\mathbf{y}\left(t_n + \frac{17h}{14}\right)^{\mathrm{T}}, \mathbf{y}\left(t_n + \frac{23h}{60}\right)^{\mathrm{T}}, \mathbf{y}(t_n + h)^{\mathrm{T}}, h\mathbf{y}'(t_n + h)^{\mathrm{T}}\right)^{\mathrm{T}}.$$

This method has step point order 3.

3. Approximate factorization iteration

In order to define the approximate factorization iteration method, we first need to extract the implicit relations to be solved from the GLM (2.5). This will be the subject of section 3.1. In section 3.2, we will specify the iteration method by using the splitting mentioned in the introduction, and in section 3.3, the computational costs of the iteration method will be discussed.

3.1. Structure of the implicit relations

To see the structure of the implicit relations to be solved, it is convenient to partition the components $\mathbf{u}_{n+1,i}$ of $\mathbf{U}_{n+1}$ into

(i) *explicit* stage values that can be explicitly evaluated by means of already computed stage values and right-hand side values, and

(ii) *implicit* stage values which need the solution of a (usually nonlinear) system of equations.

For instance, in (2.7), all stage values are explicit except for the second one, and in (2.8) and (2.9), only the first two stages are implicit and the other stages are explicit.

In most methods available in the literature, the components of $\mathbf{U}_{n+1}$ can be arranged in such a way that

$$\mathbf{U}_{n+1} = (\mathbf{X}_{n+1}^{\mathrm{T}}, \mathbf{Y}_{n+1}^{\mathrm{T}}, \mathbf{Z}_{n+1}^{\mathrm{T}})^{\mathrm{T}},$$

where $\mathbf{X}_{n+1}$ and $\mathbf{Z}_{n+1}$ represent explicit stage values and $\mathbf{Y}_{n+1}$ the implicit stage values (see again (2.7), (2.8) and (2.9)). The corresponding partitioning of the matrix T takes the form

$$T = \begin{pmatrix} L_1 & \mathrm{O} & \mathrm{O} \\ T_{21} & A & \mathrm{O} \\ T_{31} & T_{32} & L_2 \end{pmatrix}, \tag{3.1}$$

where L_1 and L_2 are strictly lower triangular matrices and T_{21}, T_{31}, T_{32} and A are allowed to be full matrices with A nonsingular. From (3.1) it follows that the implicit stage values are defined by

$$\mathbf{R}_n(\mathbf{Y}_{n+1}) = \mathbf{0}, \qquad \mathbf{R}_n(\mathbf{Y}) := \mathbf{Y} - h^2(A \otimes I)\mathbf{F}(\mathbf{Y}) - \mathbf{V}_n, \tag{3.2}$$

where $\mathbf{V}_n$ can be expressed in terms of already computed quantities. The structure of the implicit relations defining the implicit stage values is mainly determined by the matrix A. For the implicit GLMs defined by (2.6)–(2.8) and (2.9), the matrix A is respectively given by:

$$A=\widetilde{T}^2, \qquad A=\frac{4}{9}, \qquad A=\frac{1}{36}\begin{pmatrix}4 & -2\\ 18 & 0\end{pmatrix},$$
$$A=\begin{pmatrix}289/392 & 0\\ -234179/352800 & 289/392\end{pmatrix}, \tag{3.3}$$

where we assumed that in (2.6) the matrix $\widetilde{T}$ is nonsingular. In the following, the number of implicit stages will be denoted by s.

Before discussing the solution of the implicit relation (3.2), we remark that for stiff problems it is recommendable to impose a special structure on the matrices S and T such that the evaluation of *explicit* right-hand side values can be avoided. This considerably improves the accuracy in actual implementations. To be more precise, let R, S and T be partitioned according to the partitioning

$$\mathbf{U}_{n+1}=(\mathbf{X}_{n+1}^{\mathrm{T}},\mathbf{Y}_{n+1}^{\mathrm{T}},\mathbf{Z}_{n+1}^{\mathrm{T}})^{\mathrm{T}},$$

and let

$$R=\begin{pmatrix}R_1\\ R_2\\ R_3\end{pmatrix}, \qquad S=\begin{pmatrix}O & S_{12} & O\\ O & S_{22} & O\\ O & S_{32} & O\end{pmatrix}, \qquad T=\begin{pmatrix}O & O & O\\ O & A & O\\ O & T_{32} & O\end{pmatrix}, \tag{3.4}$$

where A is a nonsingular $s\times s$ matrix (note that the methods (2.7), (2.8) and (2.9) possess parameter matrices of this form). The GLM takes the form:

$$\begin{aligned}\mathbf{X}_{n+1}&=(R_1\otimes I)\mathbf{U}_n+h^2(S_{12}\otimes I)\mathbf{F}(\mathbf{Y}_n),\\ \mathbf{Y}_{n+1}&=(R_2\otimes I)\mathbf{U}_n+h^2(S_{22}\otimes I)\mathbf{F}(\mathbf{Y}_n)+h^2(A\otimes I)\mathbf{F}(\mathbf{Y}_{n+1}),\\ \mathbf{Z}_{n+1}&=(R_3\otimes I)\mathbf{U}_n+h^2(S_{32}\otimes I)\mathbf{F}(\mathbf{Y}_n)+h^2(T_{32}\otimes I)\mathbf{F}(\mathbf{Y}_{n+1}).\end{aligned}$$

Using a similar approach as used by Shampine [9] in the implementation of implicit RK methods (see also Hairer and Wanner [6, p. 129]), we express $\mathbf{F}(\mathbf{Y}_{n+1})$ in terms of $\mathbf{Y}_{n+1}$, $\mathbf{U}_n$ and $\mathbf{F}(\mathbf{Y}_n)$, i.e.,

$$h^2\mathbf{F}(\mathbf{Y}_{n+1})=\big(A^{-1}\otimes I\big)\mathbf{Y}_{n+1}-\big(A^{-1}R_2\otimes I\big)\mathbf{U}_n-h^2\big(A^{-1}S_{22}\otimes I\big)\mathbf{F}(\mathbf{Y}_n), \tag{3.5a}$$

so that we can write the GLM in the equivalent form:

$$\begin{aligned}\mathbf{X}_{n+1}&=(R_1\otimes I)\mathbf{U}_n+h^2(S_{12}\otimes I)\mathbf{F}(\mathbf{Y}_n),\\ \mathbf{Y}_{n+1}&=(R_2\otimes I)\mathbf{U}_n+h^2(S_{22}\otimes I)\mathbf{F}(\mathbf{Y}_n)+h^2(A\otimes I)\mathbf{F}(\mathbf{Y}_{n+1}),\\ \mathbf{Z}_{n+1}&=\big((R_3-T_{32}A^{-1}R_2)\otimes I\big)\mathbf{U}_n+h^2\big((S_{32}-T_{32}A^{-1}S_{22})\otimes I\big)\mathbf{F}(\mathbf{Y}_n)\\ &\quad+\big(T_{32}A^{-1}\otimes I\big)\mathbf{Y}_{n+1}.\end{aligned} \tag{3.5b}$$

Since $h^2\mathbf{F}(\mathbf{Y}_n)$ can be generated by applying (3.5a) for $n = 1, 2, \ldots, n-1$, no *explicit* $\mathbf{F}$ evaluations are needed in (3.5) except for $\mathbf{F}(\mathbf{Y}_1)$. We shall use the formulas (3.5b) in the stability analysis of the iterated GLM (see section 5.2).

3.2. *The iteration method*

Each step by the method (2.5) requires the solution of the system $\mathbf{R}_n(\mathbf{Y}) = \mathbf{0}$ specified in (3.2). In order to solve this system, we consider the modified Newton iteration process:

$$M\big(\mathbf{Y}^{(j)} - \mathbf{Y}^{(j-1)}\big) = -\mathbf{R}_n\big(\mathbf{Y}^{(j-1)}\big), \qquad M := I - A \otimes h^2 J, \quad j = 1, 2, \ldots, m, \tag{3.6a}$$

where M represents an approximation to the Jacobian matrix of $\mathbf{R}_n(\mathbf{Y})$ and $\mathbf{Y}^{(0)}$ is provided by some predictor formula. If $\mathbf{R}_n(\mathbf{Y})$ is linear, as is the case in the examples of section 2.1, then only one Newton iteration is needed, so that the solution of (3.2) is obtained by solving the linear system $M(\mathbf{Y}_{n+1} - \mathbf{Y}^{(0)}) = -\mathbf{R}_n(\mathbf{Y}^{(0)})$.

If the dimension d in the system (1.1) is large, then solving (3.6a) by *direct* methods is usually quite costly, both in computing time and in storage (see also section 3.3). In order to reduce computational costs, one may resort to *iterative* linear solvers. These linear solvers may be considered as the *inner* iteration process and the Newton process (3.6a) as the *outer* iteration process. One class of iterative solvers are the iterative preconditioned conjugate gradient methods (see, e.g., [4]). In the case of ODEs originating from PDEs in two or three space dimensions, these methods are usually much cheaper than direct methods, but they are still quite storage consuming.

It is the aim of this paper to design a storage economic, parallel iterative linear system solver that is much more efficient than the direct solution process. This solver is based on a splitting of the Jacobian of $\mathbf{f}$ into a sum of matrices J_i, so that the matrix M can be expressed as

$$M = I - A \otimes h^2 J = \frac{1}{\sigma} \sum_{i=1}^{\sigma} \big(I - \sigma A \otimes h^2 J_i\big), \tag{3.6b}$$

and on the approximate factorization of the matrix M yielding the inner–outer iteration process

$$\begin{aligned} \Pi\big(\mathbf{Y}^{(j,\nu)} - \mathbf{Y}^{(j,\nu-1)}\big) &= M\big(\mathbf{Y}^{(j-1,r)} - \mathbf{Y}^{(j,\nu-1)}\big) - \mathbf{R}_n\big(\mathbf{Y}^{(j-1,r)}\big), \\ \Pi &:= \prod_{i=\sigma}^{1} \big(I - B \otimes h^2 J_i\big), \end{aligned} \tag{3.7}$$

where $\nu = 1, 2, \ldots, r$, $j = 1, 2, \ldots, m$, $\mathbf{Y}^{(j,0)} = \mathbf{Y}^{(j-1,r)}$, and where B is a suitably chosen matrix. Evidently, if the iterates $\mathbf{Y}^{(j,\nu)}$ converge with ν, then they can only converge to the solution of (3.6a) with $\mathbf{Y}^{(j)}$ replaced by $\mathbf{Y}^{(j-1,r)}$. We will refer to (3.7) as AF iteration.

Each inner iteration in (3.7) requires the solution of σ linear systems with system matrix $I - B \otimes h^2 J_i$ of order sd. It is now clear what we meant by the preposition that the "partial" Jacobians J_i should have an "essentially simpler structure", viz. "solving the linear systems with system matrix $I - B \otimes h^2 J_i$ should be much more easy than solving the linear system in (3.6a)". For examples we refer to the spatially discretized PDEs discussed in section 2.1. It should be remarked that the idea of factorizing the system matrix of linear equations originating from multidimensional PDEs was already used in the famous ADI method of Peaceman and Rachford [8] in 1955.

The inner iteration process in (3.7) is particularly attractive if parallel computer systems are available, because the σ LU-decompositions of the system matrices $I - B \otimes h^2 J_i$ can all be done concurrently. Moreover, if B is diagonal, then the factor matrices $I - B \otimes h^2 J_i$ of the system matrix Π are block-diagonal, which enables us to decouple each of the linear systems into s subsystems which can again be solved concurrently. If B is not diagonal, but similar to a diagonal matrix with *real* diagonal entries, then we can diagonalize the iteration method (3.7) by means of a Butcher transformation $\mathbf{Y}^{(j,\nu)} = (Q \otimes I)\widetilde{\mathbf{Y}}^{(j,\nu)}$, where Q is such that $D := Q^{-1}AQ$ is diagonal (see, e.g., [6, p. 128]). Thus,

$$\widetilde{\Pi}\big(\widetilde{\mathbf{Y}}^{(j,\nu)} - \widetilde{\mathbf{Y}}^{(j,\nu-1)}\big) = -\big(Q^{-1} \otimes I\big)M(Q \otimes I)\widetilde{\mathbf{Y}}^{(j,\nu-1)} + \big(Q^{-1} \otimes I\big)\big(M\mathbf{Y}^{(j-1,r)} - \mathbf{R}_n\big(\mathbf{Y}^{(j-1,r)}\big)\big), \tag{3.7'}$$

$$\widetilde{\Pi} := \big(Q^{-1} \otimes I\big)\Pi(Q \otimes I) = \prod_{i=\sigma}^{1}\big(I - D \otimes h^2 J_i\big).$$

Evidently, the factor matrices $I - D \otimes h^2 J_i$ of the system matrix $\widetilde{\Pi}$ are again block-diagonal, allowing the same amount of parallelism as in the case where B is diagonal.

3.3. Computational costs

In order to see the typical gains in performance when using AF iteration, we consider the case where the matrix A in (3.6a) is lower triangular and the matrix B in (3.7) is diagonal.

Solving the linear Newton systems (3.6a) *directly* requires the LU-decomposition (LUD) of the s matrices $M_j := I - a_{jj}h^2 J$ $(j = 1, \ldots, s)$ and ms forward/backward substitutions (FBSs) associated with the $d \times d$ matrix M_j. Furthermore, we should also add the costs for computing m right-hand side (RHS) terms $\mathbf{R}_n(\mathbf{Y}^{(j-1)})$ in (3.6a), that is, $msdq$ flops, where q is the averaged number of flops for one scalar RHS component.

If we use a preconditioned conjugate gradient (CG) type linear solver, then the CG costs for solving the s linear subsystems are $rms\mathrm{O}(d)$ flops, where the order constant is quite large because each iteration again requires the solution of a linear system due to the preconditioning. The RHS costs are again $msdq$ flops. It is difficult to give an estimate for the number of inner iterations r. If no good initial iterate is

available, then r is at best $O(\sqrt{d})$, which happens in the case of symmetric, positive definite systems using a good preconditioner (cf. [4]). However, within one integration step, the final iterate of each inner iteration is an excellent initial iterate for the next inner iteration. Hence, it seems fair to count $rsO(d)$ flops for the CG costs instead of $rmsO(d)$ flops.

Using AF iteration (3.7) for solving the Newton systems requires the LUD of the $s\sigma$ matrices $M_{ij} := I - b_{jj}h^2J_i$ $(i = 1,\ldots,\sigma;\ j = 1,\ldots,s)$ and $rms\sigma$ FBSs associated with M_{ij}. The RHS costs in (3.7) are $msdq + rmsd(1 + \frac{1}{2}(s+1)d) \approx msd(q + \frac{1}{2}r(s+1)d)$ flops if $r > 1$ and $msdq$ flops if $r = 1$.

Usually, the LUDs needed in the direct and AF iteration methods have to be computed only every few steps (if the problem is linear, then they have to be computed only once).

Let us compare these costs in the case of a three-dimensional, semidiscrete wave equation (2.2). Then, $\sigma = 3$ and $d = N^3$, where N is the number of grid points in each direction of the spatial domain. Since this problem is linear, the CG process requires only one outer iteration ($m = 1$) and the rate of convergence of the AF process only depends on rm (see remark 4.1), that is, we may set $r = 1$, which reduces the RHS costs considerably. Furthermore, the LUDs need to be computed only once. In the flop calculation, we use the fact that the LUD and the FBS of a matrix of order n with half band width $b \ll n$ requires $2nb^2$ and $2nb$ flops, respectively (see, e.g., [4, p. 151]).

The matrix J, and therefore the matrices M_j, are banded matrices of order N^3 with half band width N^2, and the matrices M_{ij} are essentially tridiagonal matrices of order N^3. Ignoring the costs of computing the preconditioner in the CG method, the numbers of LUD, FBS, CG and RHS flops per step are given by the expressions listed in table 1 (since the numbers of outer iterations m may differ for the direct approach and the AF approach, we have denoted them m_1 and m_2, respectively).

From this table it follows that for larger values of N the total number of flops required by the direct, the CG, and AF approach are $2shN^7$, $sO(N^{9/2})$ and $s(6h + 6m_2 + m_2q)N^3$, respectively (the factor h reflects that the problem is linear, so that the

Table 1
Number of flops required in problem (2.2) with 3 spatial dimensions.

Method	p	LUD	FBS	CG	RHS
Direct approach	1	$2shN^7$	$2m_1sN^5$		m_1sN^3q
CG iteration	1			$sO(N^{9/2})$	sN^3q
AF iteration	1	$6shN^3$	$6m_2sN^3$		m_2sN^3q
Parallel direct approach	$p \geqslant 2$	$2hN^7 \max\{1, p^{-1}\}$	$2m_1sN^5$		$m_1sN^3qp^{-1}$
Parallel CG iteration	$p \geqslant 2$			$\max\{1, p^{-1}\}O(N^{9/2})$	sN^3qp^{-1}
Parallel AF iteration	$p \geqslant 2$	$2hN^3 \max\{1, 3sp^{-1}\}$	$6m_2sN^3p^{-1}$		$m_2sN^3qp^{-1}$

LUDs can be computed at the beginning of the integration process). In experiments with AF iteration applied to semidiscrete transport problems (as reported in [7]) it turned out that for $r = 1$ and $s \leqslant 2$ the implicit relations were solved with sufficient accuracy within 3 outer iterations. Since in the second-order ODE case AF iteration is expected to converge faster than in the first-order ODE case (see the discussion of the iteration error estimate (4.2) in the next section), we anticipate that $m_2 \leqslant 3$. Hence, AF iteration is expected to require at most $s(18 + 6h + 3q)N^3$ flops, which is for large N considerably less than required by the direct method and the CG method.

Next we consider the scope for parallelism of the two iteration processes on a p-processor system. In the direct approach, all LU-decompositions and all components of $\mathbf{R}_n(\mathbf{Y}^{(j-1)})$ can be computed in parallel. Hence, on a parallel computer system with $p \geqslant s$ processors, the effective LUD costs are a factor s and the effective RHS costs a factor p lower (see table 1). In the CG approach, the RHSs and the subsystem iterates can evidently be computed in parallel, but introducing parallelism in the CG iteration itself still offers difficulties. The AF approach has more scope for parallelism. Firstly, all LUDs of the matrices M_{ij} can be computed in parallel requiring $3s$ processors. Secondly, each of the systems with system matrix M_{ij} consists of N^2 tridiagonal subsystems of order N which are all uncoupled. Hence, if p processors are available, then both the FBS and RHS costs can be reduced by a factor p. For example, for $p = 3s$ and $m_2 = 3$ the total number of flops becomes effectively at most $(6 + 2h + q)N^3$.

4. Convergence results

Let us consider the behaviour of the iteration error $\varepsilon^{(j,\nu)} := \mathbf{Y}^{(j,\nu)} - \mathbf{Y}_{n+1}$. From (3.2) and (3.7) it follows that

$$\begin{aligned} \varepsilon^{(j,\nu)} &= Z\varepsilon^{(j,\nu-1)} + h^2\Pi^{-1}(A \otimes I)\mathbf{G}_n\big(\varepsilon^{(j-1,r)}\big), \quad Z := I - \Pi^{-1}M, \\ \mathbf{G}_n(\varepsilon) &:= \mathbf{F}(\mathbf{Y}_{n+1} + \varepsilon) - \mathbf{F}(\mathbf{Y}_{n+1}) - (I \otimes J)\varepsilon, \end{aligned} \tag{4.1}$$

where J is the same approximation to the Jacobian matrix as used in (3.6) and Z represents the inner amplification matrix. From the relation $\mathbf{Y}^{(j,0)} = \mathbf{Y}^{(j-1,r)}$ it follows that $\varepsilon^{(j,0)} = \varepsilon^{(j-1,r)}$, so that after r inner iterations this recursion yields

$$\varepsilon^{(j,r)} = Z^r\varepsilon^{(j-1,r)} + h^2\big(I - Z^r\big)M^{-1}(A \otimes I)\mathbf{G}_n\big(\varepsilon^{(j-1,r)}\big). \tag{4.2}$$

Let $\mathbf{G}_n$ possess a Lipschitz constant $L_n(h)$ in the neighbourhood of the origin (with respect to the norm $\|\cdot\|$) and let $L_n(h) = \mathrm{O}(h^u)$, where u depends on the Jacobian update strategy. For example, if J is updated every few steps, then $u = 1$. It is easily verified that $Z = (A - B) \otimes h^2J + \mathrm{O}(h^4)$, so that $Z = \mathrm{O}(h^\theta)$, where $\theta = 2$ if $A \neq B$ and $\theta = 4$ if $A = B$. Hence, it follows from (4.2) that

$$\big\|\varepsilon^{(j,r)}\big\| \leqslant \big(\mathrm{O}\big(h^{\theta r}\big) + \mathrm{O}\big(h^{u+2}\big)\big)\big\|\varepsilon^{(j-1,r)}\big\|, \quad j \geqslant 1. \tag{4.2'}$$

This estimate shows that we have at least fast convergence of the nonstiff components. For example, if $u = 1$, then in each outer iteration the iteration error is damped by a

factor $O(h^{\theta r}) + O(h^3)$. Hence, choosing $r = 4\theta^{-1}$, we may expect a convergence rate comparable with that of modified Newton. We remark that application of AF iteration to first-order ODE methods (see [3]) yields an inner amplification matrix Z satisfying $Z = O(h^{\theta/2})$, so that we may expect faster convergence in the second-order ODE case.

Remark 4.1. If the ODE is *linear*, then $\mathbf{G}_n$ vanishes, so that by means of (4.2) $\varepsilon^{(j,r)} = Z^r \varepsilon^{(j-1,r)} = Z^{rj} \varepsilon^{(0,r)}$, and therefore, $\varepsilon^{(m,r)} = Z^{rm} \varepsilon^{(0,r)}$. This shows that for linear problems, the rate of convergence of AF iteration only depends on Z^{rm}, that is, only on the product rm. This means that in linear problems we should set $r = 1$ by which the expensive matrix–vector multiplication in (3.7) is avoided (of course, the value of m is expected to be larger than in the direct approach).

So far, all our considerations were independent of the splitting of the Jacobian J. However, in the remainder of this section, we will focus on the convergence in the case of model problems.

4.1. The model problem

The case where the "partial" Jacobians J_i all commute with each other, that is, they share the same eigensystem, will be referred to as the *model problem*. Such model situations occur if (1.1) originates from certain classes of second-order partial differential equations (see section 2.1).

For briefness of notation, we introduce the following convention. Let $E(h^2 J_1, \ldots, h^2 J_\sigma)$ be a matrix depending on $h^2 J_1, \ldots, h^2 J_\sigma$. Then the $s \times s$ matrix obtained by replacing the matrices $h^2 J_i$ by the scalars z_i is denoted by $E(\mathbf{z})$, where $\mathbf{z} = (z_1, \ldots, z_\sigma)$. Thus, with the matrices M defined in (3.6b), Π defined in (3.7), and Z defined in (4.2) we associate the matrices

$$Z(\mathbf{z}) := I - \Pi^{-1}(\mathbf{z}) M(\mathbf{z}), \qquad M(\mathbf{z}) = I - (\mathbf{e}^{\mathrm{T}}\mathbf{z}) A, \qquad \Pi(\mathbf{z}) = \prod_{i=\sigma}^{1} (I - z_i B), \quad (4.3)$$

where $\mathbf{e}$ is the σ-dimensional vector with unit entries. Evidently, if we choose $z_i := \lambda(J_i) h^2$, where $\lambda(J_i)$ denotes an eigenvalue of J_i, then in the case of the model problem defined above, the eigenvalues of the amplification matrix in (4.1) are given by those of the matrix $Z(\mathbf{z})$. The region of convergence can then be defined by the region in the $\mathbf{z}$-plane where $Z(\mathbf{z})$ has its eigenvalues $\zeta(\mathbf{z})$ within the unit circle. Assuming that the eigenvalues of the "partial" Jacobians J_i are on the *negative real axis* (as is the case in many wave equation problems), we shall call the iteration method (3.7) *A(0)-convergent* if the region of convergence contains the region $\{\mathbf{z}: z_i \leqslant 0\}$. The eigenvalues $\zeta(\mathbf{z})$, will be called the *amplification factors* of the inner iteration method.

4.2. Matrices $B = A$ with real eigenvalues

We consider the convergence region of (3.7) in the case where $B = A$ with $\lambda(A)$ real (for example, as in the methods (2.7) and (2.9)). The amplification factors are given by

$$\zeta(\mathbf{z}) := 1 - \pi^{-1}(\mathbf{z})\mu(\mathbf{z}), \quad \mu(\mathbf{z}) := 1 - \lambda(A)\big(\mathbf{e}^{\mathrm{T}}\mathbf{z}\big), \quad \pi(\mathbf{z}) := \prod_{i=1}^{\sigma}\big(1 - \lambda(A)z_i\big), \tag{4.4}$$

where $\lambda(A)$ denotes an eigenvalue of A. Let $\lambda(A) \geqslant 0$. Then, it follows from (4.4) that $A(0)$-convergence is achieved if $2\pi(\mathbf{z}) - \mu(\mathbf{z}) > 0$ for $z_i \leqslant 0$. Since we may write

$$\pi(\mathbf{z}) = \mu(\mathbf{z}) + p_2\lambda^2(A) + p_3\lambda^3(A) + \cdots + p_\sigma\lambda^\sigma(A),$$

where the coefficients p_i are non-negative whenever $z_i \leqslant 0$, we see that for $\lambda(A) \geqslant 0$ and $z_i \leqslant 0$

$$2\pi(\mathbf{z}) - \mu(\mathbf{z}) = \mu(\mathbf{z}) + 2\big(p_2\lambda^2(A) + p_3\lambda^3(A) + \cdots + p_\sigma\lambda^\sigma(A)\big) > 0.$$

Theorem 4.1. If $\lambda(A) \geqslant 0$, then AF iteration $\{(3.7),\ B = A\}$ is $A(0)$-convergent for all σ.

4.3. Matrices $B = A$ with complex eigenvalues

If $B = A$ with A having complex eigenvalues, then the convergence analysis is more complicated. We separately discuss the cases of two and three splitting terms ($\sigma = 2$ and $\sigma = 3$).

4.3.1. Two splitting terms

If $\sigma = 2$, then the amplification factor can be factorized according to

$$\zeta(\mathbf{z}) = \lambda(A)z_1\big(1 - \lambda(A)z_1\big)^{-1}\lambda(A)z_2\big(1 - \lambda(A)z_2\big)^{-1}. \tag{4.5}$$

By requiring that the magnitude of both factors is less than 1, we see that for $\sigma = 2$ the region of convergence of the inner iteration method in (3.7) contains the domain

$$\mathbb{D} := \bigcap_{\lambda(A)}\left\{\mathbf{z}\colon\ z_j\,\mathrm{Re}\big(\lambda(A)\big) < \frac{1}{2},\ j = 1,2\right\}.$$

Theorem 4.2. If $\mathrm{Re}(\lambda(A)) \geqslant 0$, then AF iteration $\{(3.7),\ B = A\}$ is $A(0)$-convergent for $\sigma = 2$.

Thus, AF iteration applied to (2.7), (2.8) and (2.9) is $A(0)$-convergent. In the particular case of the indirect GLM (2.6), we immediately have by virtue of theorem 4.2 the result:

Corollary 4.1. If the generating GLM $(\widetilde{\mathbf{a}}, \widetilde{R}, \widetilde{S}, \widetilde{T})$ in the indirect GLM (2.6) satisfies $|\arg(\lambda(\widetilde{T}))| \leqslant \pi/4$, then AF iteration $\{(3.7),\ A = B = \widetilde{T}^2\}$ is $A(0)$-convergent for $\sigma = 2$.

This corollary implies that for all indirect RKN methods generated by RK matrices whose Butcher matrices $\widetilde{A}$ have their eigenvalues in the wedge $|\arg(\lambda(\widetilde{A}))| \leqslant \pi/4$ AF iteration is $A(0)$-convergent. For example, this happens in the case of the third-order Radau IIA, the fourth-order Lobatto IIIA and the fourth-order and sixth-order Gauss methods.

Next, consider the case where A has eigenvalues with $\mathrm{Re}(\lambda(A)) < 0$, so that $A(0)$-convergence is not possible. In fact, the region of convergence consists of two strips along the negative z_1-axis and the negative z_2-axis. The plot in figure 1 is typical for the form of the region of divergence in the third quadrant of the (z_1, z_2)-plane obtained for methods with $\mathrm{Re}(\lambda(A)) < 0$ (gray part indicates divergence). Note that the convergence region is symmetric with respect to the line $z_1 = z_2$.

In a number of important applications, we do not need $A(0)$-convergence with respect to both z_1 and z_2. For example, in the 2-dimensional modeling of the water elevation in a river, we encounter a wave equation in which the resolution of the coordinate perpendicular to the river should be an order of magnitude smaller than the resolution of the coordinate along the river. Hence, the "stiffness" of the Newton systems (3.6a) comes from the direction perpendicular to the river, so that we need only unconditional convergence with respect to this direction. In such cases, a region of convergence as in figure 1 is quite sufficient.

If we have stiffness with respect to both z_1 and z_2, then we should look at the disk, centered at the origin, which is contained in the region of convergence. From figure 1 it follows that the radius of this disk can be determined by setting $z_1 = z_2$ on the boundary of the convergence region. Hence, the point $z_0\mathbf{e}$ is on the boundary

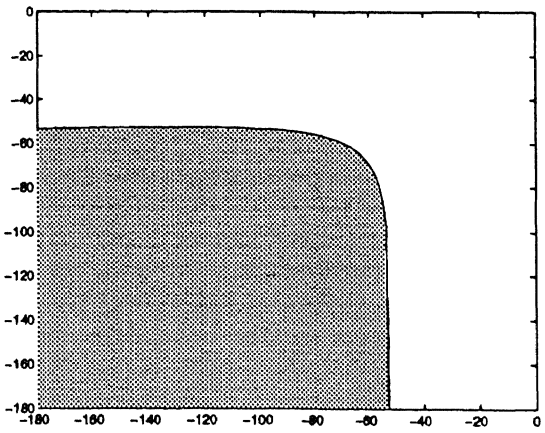


Figure 1. Divergence region for AF iteration applied to the RKN method generated by fifth-order Radau IIA method (example 4.1) with $A = B$.

of this convergence disk if z_0 is a solution (nearest to the origin) of the equations $|\zeta(z_0\mathbf{e}))| = 1$ associated with those eigenvalues $\lambda(A)$ of A that are in the negative halfplane. From (4.5) it follows that z_0 satisfies $|\lambda(A)z_0(1 - \lambda(A)z_0)^{-1}| = 1$. This equation has just one solution given by $[2\mathrm{Re}(\lambda(A))]^{-1}$, so that we may conclude that the convergence region of the inner iteration method in (3.7) contains the domain

$$\mathbb{D} = \left\{ \mathbf{z}:\ z_1^2 + z_2^2 < 2z_0^2,\ z_0 := \max_{\mathrm{Re}(\lambda(A))<0} \frac{1}{2\mathrm{Re}(\lambda(A))},\ z_1 \leqslant 0,\ z_2 \leqslant 0 \right\}. \tag{4.6}$$

Suppose that the matrices J_1 and J_2 possess the spectral radius $\rho(J_1)$ and $\rho(J_2)$. Then the convergence condition becomes $h^4(\rho^2(J_1) + \rho^2(J_2)) < 2z_0^2$. Thus, we have the convergence result:

Theorem 4.3. Let $\sigma = 2$. If A has one or more eigenvalues in the negative halfplane, then a sufficient condition for convergence of AF iteration $\{(3.7),\ B = A\}$ is given by

$$h < \left(\frac{2z_0^2}{\rho^2(J_1) + \rho^2(J_2)} \right)^{1/4}, \quad z_0 := \max_{\mathrm{Re}(\lambda(A))<0} \frac{1}{2\mathrm{Re}(\lambda(A))}. \tag{4.7}$$

Example 4.1. We illustrate this convergence result by means of the RKN method generated by the fifth-order Radau IIA method for first-order ODE methods. From (2.6) it follows that the RKN matrix A is the square of the Radau IIA matrix, so that

$$\begin{aligned} A &= \begin{pmatrix} \frac{88 - 7\sqrt{6}}{360} & \frac{296 - 169\sqrt{6}}{1800} & \frac{-2 + 3\sqrt{6}}{225} \\ \frac{296 + 169\sqrt{6}}{1800} & \frac{88 + 7\sqrt{6}}{360} & \frac{-2 - 3\sqrt{6}}{225} \\ \frac{16 - \sqrt{6}}{36} & \frac{16 + \sqrt{6}}{36} & \frac{1}{9} \end{pmatrix}^2 \\ &\approx \begin{pmatrix} 0.022 & -0.020 & 0.010 \\ 0.177 & 0.038 & -0.007 \\ 0.318 & 0.182 & 0.000 \end{pmatrix}. \end{aligned} \tag{4.8}$$

Its eigenvalues are given by $\lambda(A) \approx 0.0756$ and $\lambda(A) \approx -0.0078 \pm 0.0601\,\mathrm{i}$. Applying theorem 4.3 results in the convergence condition $h < 9.52[\rho^2(J_1) + \rho^2(J_2)]^{-1/4}$.

When A has one or more eigenvalues in the left halfplane, one may wonder whether the fixed point iteration (FP) process might be a better approach than the AF process. To answer this question, we should consider the FP error equation. By observing that using FP iteration for solving the Newton systems in (3.6a) yields an inner–outer iteration process of the form (3.7) with $B = \mathrm{O}$, i.e., $\Pi = I$, the inner amplification matrix Z reduces to

$$Z = I - M = A \otimes h^2 J. \tag{4.9}$$

This relation shows that FP iteration converges if $h < [\rho(J)\rho(A)]^{-1/2}$. A comparison with (4.7) yields:

Theorem 4.4. Let $\sigma = 2$. If A has one or more eigenvalues in the negative halfplane, then the interval of convergent stepsizes h of AF iteration $\{(3.7),\ B = A\}$ is larger than that of FP iteration $\{(3.7),\ B = O\}$ if

$$\frac{\rho^2(J_1) + \rho^2(J_2)}{\rho^2(J_1 + J_2)} < \frac{1}{2} \min_{\mathrm{Re}(\lambda(A))<0} \left(\frac{\rho(A)}{\mathrm{Re}(\lambda(A))} \right)^2. \tag{4.10}$$

For example, if we use a splitting according to dimensions in the 2-dimensional wave equation, then $\rho(J_1) = \rho(J_2) = \rho(J_1 + J_2)/2$, so that the left-hand side of (4.10) becomes $1/2$. Hence, (4.10) is always satisfied.

There are, of course, other aspects that should be taken into account. AF iteration needs LU decompositions and forward–backward substitutions. On the other hand, the amplification factor is much better for AF iteration. In order to appreciate the damping of the initial error $\mathbf{Y}^{(0)} - \mathbf{Y}_{n+1}$ by the two iteration methods, we compare the amplification factor (4.5) with the amplification factor associated with (4.9). For the AF method, the largest amplification factors occur on the line $z_1 = z_2 = z/2$, so that along this line their magnitudes are respectively given by

$$\zeta_{\mathrm{AF}} = \max_{\mathrm{Re}(\lambda(A))<0} \frac{|\lambda(A)|^2 z^2}{4 - 4\mathrm{Re}(\lambda(A))z + |\lambda(A)|^2 z^2}, \quad \zeta_{\mathrm{FP}} = \left|\rho(A)z\right|, \quad z := h^2\lambda(J).$$

An important aspect is that ζ_{AF} increases only slightly beyond 1, so that using too large stepsizes never causes a violent divergence behaviour as would be the case when FP iteration is applied. In fact, ζ_{AF} will never exceed the value $(1-[\mathrm{Re}(\lambda(A))|\lambda(A)|^{-1}]^2)^{-1}$. For example, in the case of the fifth-order Radau IIA-based method (4.9), this maximal value is about 1.017.

4.3.2. Three splitting terms

For three splitting terms ($\sigma = 3$) we can obtain a spectrum condition on A by using the following lemma (for a proof see [3]):

Lemma 4.1. Let $\mathbf{w} := (w_1, w_2, w_3)$ and define the functions $p(\mathbf{w}) := (1 - w_1)(1 - w_2)(1 - w_3)$ and $m(\mathbf{w}) := 1 - \mathbf{e}^{\mathrm{T}}\mathbf{w}$, where w_j are complex variables. Then, in the region $\{\mathbf{w}:\ 3\pi/4 \leqslant \arg(w_j) \leqslant 5\pi/4\}$, the function $1 - p^{-1}(\mathbf{w})m(\mathbf{w})$ assumes values within the unit circle.

From (4.4) it follows that $\zeta(\mathbf{z}) = 1 - p^{-1}(\lambda(A)\mathbf{z})m(\lambda(A)\mathbf{z})$. Applying lemma 4.1 with $w_j = \lambda(A)z_j$, we see that $\zeta(\mathbf{z})$ assumes values within the unit circle in the region $\{\mathbf{z}:\ 3\pi/4 \leqslant \arg(\lambda(A)z_j) \leqslant 5\pi/4\}$. Thus, we have the result:

Theorem 4.5. If A has eigenvalues $\lambda(A)$ with $|\arg(\lambda(A))| \leqslant \pi/4$, then AF iteration $\{(3.7),\ B = A\}$ is $A(0)$-convergent for $\sigma = 3$.

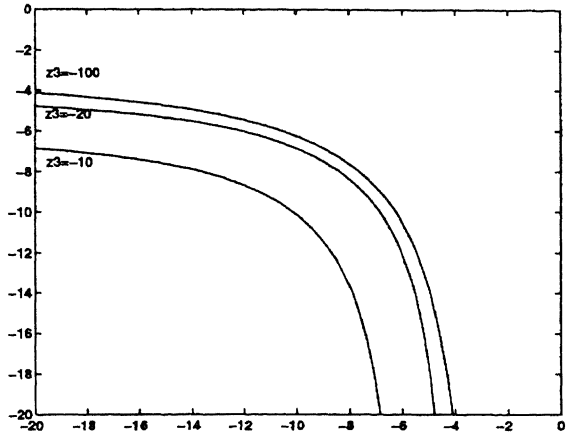


Figure 2. Convergence boundaries in the (z_1, z_2)-plane for AF iteration applied to the RKN method generated by third-order Radau IIA method (2.8) with $B = A$.

Corollary 4.2. If the generating GLM $(\widetilde{\mathbf{a}}, \widetilde{R}, \widetilde{S}, \widetilde{T})$ in the indirect GLM (2.6) satisfies $|\arg(\lambda(\widetilde{T}))| \leqslant \pi/8$, then AF iteration $\{(3.7),\ A = B = \widetilde{T}^2\}$ is $A(0)$-convergent for $\sigma = 3$.

Hence, AF iteration applied to (2.7) and (2.9) is also $A(0)$-convergent for $\sigma = 3$. However, this is not the case for the RKN methods generated by the Radau IIA, Lobatto IIIA and Gauss methods, because they all have $|\arg(\lambda(\widetilde{A}))| > \pi/8$.

If $|\arg(\lambda(A))| > \pi/4$, then the convergence region is finite and the region of divergence is a sort of hyperboloid. In order to get some idea of the region of convergence, we plotted in figure 2 for (2.8) the convergence boundaries in the (z_1, z_2)-plane for a few values of z_3.

By virtue of the symmetry with respect to z_j, the convergence region contains the domain (cf. (4.7))

$$\mathbb{D} := \bigcap_{\lambda(A)} \{\mathbf{z}:\ z_1^2 + z_2^2 + z_3^2 < 3z_0^2,\ z_j \leqslant 0,\ j = 1,2,3\},$$

where z_0 is the negative root of smallest magnitude of the equation $|1 - \pi^{-1}(z_0\mathbf{e})\mu(z_0\mathbf{e})| = 1$, that is, of the equation $|\pi(z_0\mathbf{e})|^2 - |\pi(z_0\mathbf{e}) - \mu(z_0\mathbf{e})|^2 = 0$. Let us write $\lambda(A) = r\exp(\mathrm{i}\alpha)$, and define $q := |z_0 r|$. Then it can be shown that this equation yields the following relation between q and α:

$$[1 + 3q^2 - 6q^4] + 6q[1 + 2q^2]\cos(\alpha) + 4q^2[3 - 2q + 3q^2]\cos^2(\alpha) = 0,$$
$$q \geqslant 0,\ \alpha \geqslant \pi/4. \tag{4.11}$$

This relation yields $q = \infty$ at $\alpha = \pi/4$, then rapidly decreases to ≈ 0.85 at $\alpha = \pi/2$, and slowly decreases to $q \approx 0.33$ at $\alpha = \pi$. We have the following analogue of theorem 4.3:

Theorem 4.6. Let $q = q(\alpha)$ be defined by (4.11). Then, for $\sigma = 3$ a sufficient condition for convergence of AF iteration $\{(3.7),\ B = A\}$ is given by

$$h < \left(\frac{3z_0^2}{\rho^2(J_1) + \rho^2(J_2) + \rho^2(J_3)} \right)^{1/4}, \quad z_0 := -\min_{\lambda(A)} \frac{q(\arg(A))}{|\lambda(A)|}. \tag{4.12}$$

4.4. Matrices $B \neq A$

In this section, we investigate whether the severe conditions on the spectrum of the matrix A to achieve $A(0)$-convergence derived in the preceding section 4.3 can be relaxed by choosing $B \neq A$. Some insight can be obtained by looking at the behaviour of the amplification matrix $Z(\mathbf{z})$ at infinity. We respectively consider $Z(\mathbf{z})$ in the cases where $z_i \to \infty$ and $z_j = 0$ for $j \neq i$, and in the case where all components z_i tend to infinity. This yields, respectively,

$$\begin{aligned} &Z(\mathbf{z}) \approx I - B^{-1}A + z_i^{-1}B^{-1}(I - B^{-1}A), \qquad Z(\mathbf{z}) \approx I - \delta B^{-\sigma}A, \\ &\delta := \frac{(-1)^{\sigma+1}(\mathbf{e}^{\mathrm{T}}\mathbf{z})}{z_1 \cdots z_\sigma} \quad \text{as } z_i \to \infty,\ i = 1, \ldots, \sigma. \end{aligned} \tag{4.13}$$

Since $z_i < 0$ and $\delta > 0$ we easily derive from (4.13) the following result:

Theorem 4.7. For $\sigma \geqslant 2$, the conditions $|\lambda(I - B^{-1}A)| \leqslant 1$ and $\mathrm{Re}(\lambda(B^{-\sigma}A)) \geqslant 0$ are necessary for the $A(0)$-convergence of AF iteration.

This theorem provides a guideline for choosing the matrix B.

Example 4.2. Consider the method (2.8). The eigenvalues of the matrix A are given by $\lambda(A) \approx 0.0556 \pm 0.1571\,\mathrm{i}$, so that $|\arg(\lambda(A))| \approx 0.39\pi$. If we would have chosen $B = A$, then the first necessary $A(0)$-convergence condition of theorem 4.7 is trivially satisfied. However, since $|\arg(\lambda(B^{-\sigma}A))| = |\arg(\lambda(A^{1-\sigma}))| = (\sigma - 1)|\arg(\lambda(A))| \approx 0.39(\sigma - 1)\pi$, the second condition of this theorem is violated if $0.39(\sigma - 1)\pi > \pi/2$, i.e., if $\sigma > 2.28$.

Now, let us choose B diagonal and such that $I - B^{-1}A$ has two zero eigenvalues, so that the first condition of theorem 3.6 is satisfied. This leads to

$$B = \frac{1}{18}\begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix}. \tag{4.14}$$

A straightforward calculation reveals that

$$B^{-\sigma}A = 18^{\sigma-1}\begin{pmatrix} 2 & -1 \\ 9^{1-\sigma} & 0 \end{pmatrix}, \qquad \lambda(B^{-\sigma}A) = 18^{\sigma-1}\left(1 \pm \sqrt{1 - 9^{1-\sigma}}\right), \tag{4.15}$$

so that the second condition of theorem 4.7 is also satisfied, irrespective of the value of σ. For $\sigma = 2$ and $\sigma = 3$, we checked the $A(0)$-convergence in the case where B is defined by (4.14) and verified that in both cases we have $A(0)$-convergence.

5. Fixed numbers of inner and outer iterations

If the implicit relations (3.2) are iterated until convergence, then we may rely on the order of accuracy and the stability of the underlying GLM (2.5). However, in actual computation, it is often more efficient if we do *not* iterate the outer and inner iteration process until convergence. Consequently, the order of accuracy and the stability properties of the resulting integration scheme will not be identical to those of the underlying integration method. On the other hand, there is no need for convergence of the AF iteration process.

5.1. Order of accuracy

Let us consider the order of accuracy of the step values produced by the iterated method for fixed m and r (we recall that a step value is an external stage value corresponding to a step point t_{n+1}). Let $\mathbf{u}_{n+1,i}$ be a step value in the underlying method (2.5) and let $\mathbf{u}_{n+1,i}^{(m,r)}$ be the approximation after m outer and r inner iterations. If $\mathbf{u}_{n+1,i}$ has local error of order $p+1$, then

$$\begin{aligned}\mathbf{u}_{n+1,i}^{(m,r)} - \mathbf{y}(t_n + h) &= \mathbf{u}_{n+1,i}^{(m,r)} - \mathbf{u}_{n+1,i} + \mathbf{u}_{n+1,i} - \mathbf{y}(t_{n+1}) \\ &= \mathbf{u}_{n+1,i}^{(m,r)} - \mathbf{u}_{n+1,i} + \mathrm{O}\big(h^{p+1}\big), \end{aligned} \tag{5.1}$$

where $\mathbf{y}(t_{n+1})$ denotes the locally exact solution. By observing that no iteration errors are introduced in the computation of the explicit stages, we can derive the order in h of $\mathbf{u}_{n+1,i}^{(m,r)} - \mathbf{u}_{n+1,i}$ by using the iteration error estimate (4.2′). Let the predictor for the implicit stage values have local error of order $q+1$, i.e., $\varepsilon^{(0,r)} = \mathrm{O}(h^{q+1})$. Then (5.1) and (4.2′) yield

$$\big\|\mathbf{u}_{n+1,i}^{(m,r)} - \mathbf{y}(t_n + h)\big\| = h^{q+1}\big(\mathrm{O}\big(h^{\theta r}\big) + \mathrm{O}\big(h^{u+2}\big)\big)^m + \mathrm{O}\big(h^{p+1}\big). \tag{5.2}$$

This leads us to the following result:

Theorem 5.1. Let the step point values of the underlying GLM be of order p, let the predictor formula have order q, let the function $\mathbf{G}_n$ defined in (4.1) possess a Lipschitz constant $L_n(h) = \mathrm{O}(h^u)$, and let the inner amplification matrix Z defined in (4.1) satisfy $Z = \mathrm{O}(h^\theta)$. Then the maximal order of accuracy is reached if $m \geqslant (p-q)/\min\{\theta r, u+2\}$.

Example 5.1. Let $r = 1$ (by which the RHS costs are more than proportionally minimized, see section 3.3), $\theta = 2$ (corresponding with AF iteration using $B \neq A$), and $u = 1$ (Jacobian update every few steps). Then, the number of outer iterations should not be less than $\frac{1}{2}(p-q)$. Thus, we need only one inner and one outer iteration if the order q of the predictor satisfies $q \geqslant p-2$, that is, if $p = 2$ we may use the trivial zero-order predictor $\mathbf{Y}^{(0,r)} = \mathbf{Y}_n$, and if $p = 3$ we may use a linear first-order extrapolation predictor.

5.2. *Stability*

In order to see the effect of the number of iterations on the stability, we apply the integration process to the stability test equation $\mathbf{y}' = \mathbf{J}\mathbf{y}$. We shall confine our considerations to the case where S and T have the structure as specified in (3.4), so that the GLM can be written in the form (3.5b).

Since the test equation is linear, we may set $\mathbf{G}_n = \mathbf{0}$ in (4.1). From (4.2) and (3.5b) it follows that

$$\mathbf{Y}^{(m,r)} - \mathbf{Y}_{n+1} = Z^{rm}\big(\mathbf{Y}^{(0,r)} - \mathbf{Y}_{n+1}\big),$$
$$\mathbf{Y}_{n+1} = M^{-1}\big((R_2 \otimes I)\mathbf{U}_n + \big(S_{22} \otimes h^2 J\big)\mathbf{Y}_n\big).$$

Let the predictor for the outer iteration process be given by $\mathbf{Y}^{(0,r)} = P\mathbf{U}_n$. Then,

$$\mathbf{Y}^{(m,r)} = Z^{rm}P\mathbf{U}_n + \big(I - Z^{rm}\big)M^{-1}\big((R_2 \otimes I)\mathbf{U}_n + \big(S_{22} \otimes h^2 J\big)\mathbf{Y}_n\big). \tag{5.3}$$

By identifying $\mathbf{Y}_{n+1}$ with $\mathbf{Y}^{(m,r)}$ it follows from (3.5b) that for the stability test equation

$$\begin{aligned}
\mathbf{X}_{n+1} &= (R_1 \otimes I)\mathbf{U}_n + \big(S_{12} \otimes h^2 J\big)\mathbf{Y}_n,\\
\mathbf{Y}_{n+1} &= Z^{rm}P\mathbf{U}_n + \big(I - Z^{rm}\big)M^{-1}\big[(R_2 \otimes I)\mathbf{U}_n + \big(S_{22} \otimes h^2 J\big)\mathbf{Y}_n\big],\\
\mathbf{Z}_{n+1} - \big(T_{32}A^{-1} \otimes I\big)\mathbf{Y}_{n+1} &= \big(\big(R_3 - T_{32}A^{-1}R_2\big) \otimes I\big)\mathbf{U}_n\\
&\quad + \big(\big(S_{32} - T_{32}A^{-1}S_{22}\big) \otimes h^2 J\big)\mathbf{Y}_n.
\end{aligned} \tag{5.4}$$

Thus, we obtain a relation of the type $\mathbf{U}_{n+1} = \Sigma_{mr}\mathbf{U}_n$, where Σ_{mr} is a matrix defined by (5.4) and which depends on the matrices $h^2 J_i$. Its eigenvalues are given by the eigenvalues of the matrix $\Sigma_{mr}(\mathbf{z})$, where $\Sigma_{mr}(\mathbf{z})$ is defined in the same way as the matrices $M(\mathbf{z})$, $\Pi(\mathbf{z})$ and $Z(\mathbf{z})$ in (4.3). Next we observe that due to possible internal stages, the matrix $\Sigma_{mr}(\mathbf{z})$ may contain a number of zero columns. As a consequence, the corresponding components of $\mathbf{U}_{n+1}$ do not play a role in the propagation of perturbations through the steps. Let all ith columns of $\Sigma_{mr}(\mathbf{z})$ with $i \in \mathbb{I}$ be a zero column, and let $\widetilde{\Sigma}_{mr}(\mathbf{z})$ denote the matrix obtained by removing all ith columns and ith rows from $\Sigma_{mr}(\mathbf{z})$ for $i \in \mathbb{I}$. Then, we have stability if the *stability matrix* $\widetilde{\Sigma}_{mr}(\mathbf{z})$ has its eigenvalues on the unit disk. The region of stability is defined by the region in the $\mathbf{z}$-plane where $\widetilde{\Sigma}_{mr}(\mathbf{z})$ has its eigenvalues within the unit circle (cf. the definition of the region of convergence in section 3). Again assuming that the eigenvalues of the "partial" Jacobians J_i are on the nonpositive real axis, we shall call the integration method $A(0)$-*stable* if the region of stability contains the region $\{\mathbf{z}\colon\ z_i \leqslant 0\}$.

We illustrate the above procedure by deriving the stability matrix for iterated RKN methods with step point value $\mathbf{y}_{n+1} = (\mathbf{e}_s^{\mathrm{T}} \otimes I)\mathbf{Y}_{n+1}$ and with only one explicit derivative stage value $h\mathbf{y}'_{n+1}$, i.e., $\mathbf{U}_{n+1} = (\mathbf{Y}_{n+1}^{\mathrm{T}}, h\mathbf{y}'^{\mathrm{T}}_{n+1})^{\mathrm{T}}$. Using the "last step value" predictor $\mathbf{Y}^{(0,r)} = P\mathbf{U}_n = (\mathbf{e}\mathbf{e}_s^{\mathrm{T}} \otimes I)\mathbf{Y}_n$, we have

$$R = \begin{pmatrix} \mathbf{e}\mathbf{e}_s^{\mathrm{T}} & \mathbf{c} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{pmatrix}, \qquad S = \mathrm{O}, \qquad T = \begin{pmatrix} A & \mathbf{0} \\ \mathbf{d}^{\mathrm{T}} & 0 \end{pmatrix}, \tag{5.5}$$

where $\mathbf{c}$ and $\mathbf{d}$ are s-dimensional vectors. Equations (5.4) take the form

$$\begin{aligned}
&\mathbf{Y}_{n+1} = Z^{rm}(\mathbf{e}\mathbf{e}_s^{\mathrm{T}} \otimes I)\mathbf{Y}_n + (I - Z^{rm})M^{-1}\big((\mathbf{e}\mathbf{e}_s^{\mathrm{T}} \otimes I)\mathbf{Y}_n + h(c \otimes I)\mathbf{y}'_n\big),\\
&h\mathbf{y}'_{n+1} - (\mathbf{d}^{\mathrm{T}}A^{-1} \otimes I)\mathbf{Y}_{n+1} = \big((-\mathbf{d}^{\mathrm{T}}A^{-1}\mathbf{e}\mathbf{e}_s^{\mathrm{T}}) \otimes I\big)\mathbf{Y}_n \qquad (5.4')\\
&\qquad\qquad + h\big(\big(1 - \mathbf{d}^{\mathrm{T}}A^{-1}\mathbf{c}\big) \otimes I\big)\mathbf{y}'_n.
\end{aligned}$$

Using $\mathbf{y}_{n+1} = (\mathbf{e}_s^{\mathrm{T}} \otimes I)\mathbf{Y}_{n+1}$, we obtain

$$\begin{aligned}
&\mathbf{Y}_{n+1} = \big((I - Z^{rm})M^{-1} + Z^{rm}\big)(\mathbf{e} \otimes I)\mathbf{y}_n + h(I - Z^{rm})M^{-1}(\mathbf{c} \otimes I)\mathbf{y}'_n,\\
&\mathbf{y}_{n+1} = (\mathbf{e}_s^{\mathrm{T}} \otimes I)\big((I - Z^{rm})M^{-1} + Z^{rm}\big)(\mathbf{e} \otimes I)\mathbf{y}_n\\
&\qquad + h(\mathbf{e}_s^{\mathrm{T}} \otimes I)(I - Z^{rm})M^{-1}(\mathbf{c} \otimes I)\mathbf{y}'_n, \qquad (5.4'')\\
&h\mathbf{y}'_{n+1} - (\mathbf{d}^{\mathrm{T}}A^{-1} \otimes I)\mathbf{Y}_{n+1} = -(\mathbf{d}^{\mathrm{T}}A^{-1}\mathbf{e} \otimes I)\mathbf{y}_n + h\big(\big(1 - \mathbf{d}^{\mathrm{T}}A^{-1}\mathbf{c}\big) \otimes I\big)\mathbf{y}'_n.
\end{aligned}$$

Elimination of $\mathbf{Y}_{n+1}$ leads to the 2×2 stability matrix

$$\widetilde{\Sigma}_{mr}(\mathbf{z}) = \begin{pmatrix} \mathbf{e}_s^{\mathrm{T}}\big(S_{mr}(\mathbf{z}) + Z^{mr}(\mathbf{z})\big)\mathbf{e} & \mathbf{e}_s^{\mathrm{T}}S_{mr}(\mathbf{z})\mathbf{c} \\ \mathbf{d}^{\mathrm{T}}A^{-1}\big(S_{mr}(\mathbf{z}) + Z^{mr}(\mathbf{z}) - I\big)\mathbf{e} & 1 + \mathbf{d}^{\mathrm{T}}A^{-1}(S_{mr}(\mathbf{z}) - I)\mathbf{c} \end{pmatrix}, \qquad (5.6)$$

where $S_{mr}(\mathbf{z}) := (I - Z^{mr}(\mathbf{z}))M^{-1}(\mathbf{z})$. It is of interest to study the behaviour of the stability matrix $\widetilde{\Sigma}_{mr}(\mathbf{z})$ at infinity. We consider $\widetilde{\Sigma}_{mr}(\mathbf{z})$ in the cases where $z_i \to \infty$ and $z_j = 0$ for $j \neq i$, and in the case where all components z_i tend to infinity. From relations (4.13) and

$$\begin{aligned}
&M^{-1}(\mathbf{z}) \approx z_i^{-1}A^{-1}, \qquad S_{mr}(\mathbf{z}) \approx \mathrm{O}\big(z_i^{-1}\big) \quad \text{as } z_i \to \infty,\ i = 1, \ldots, \sigma,\\
&M^{-1}(\mathbf{z}) \approx \varepsilon A^{-1}, \qquad S_{mr}(\mathbf{z}) \approx \mathrm{O}(\delta\varepsilon) \quad \text{as } \varepsilon, \delta \to 0,
\end{aligned}$$

where $\varepsilon := -(\mathbf{e}^{\mathrm{T}}\mathbf{z})^{-1}$ and δ is defined in (4.13), it follows that the two eigenvalues of $\widetilde{\Sigma}_{mr}(\mathbf{z})$ approach the values $\{\mathbf{e}_s^{\mathrm{T}}(I - B^{-1}A)^{mr}\mathbf{e},\ 1 - \mathbf{d}^{\mathrm{T}}A^{-1}\mathbf{c}\}$ and $\{1 - mr\delta(\mathbf{e}_s^{\mathrm{T}}B^{-\sigma}A\mathbf{e}),\ 1 - \mathbf{d}^{\mathrm{T}}A^{-1}\mathbf{c}\}$, respectively. Since $|1 - \mathbf{d}^{\mathrm{T}}A^{-1}\mathbf{c}| \leqslant 1$ is also needed for the $A(0)$-stability of the underlying RKN method, we have:

Theorem 5.2. Let the underlying GLM (2.5) be an $A(0)$-stable RKN method defined by (5.5) and let the initial iterate for AF iteration be defined by $\mathbf{Y}^{(0,r)} = (\mathbf{e}\mathbf{e}_s^{\mathrm{T}} \otimes I)\mathbf{Y}_n$. Then, after m outer and r inner iterations, the two conditions $|\mathbf{e}_s^{\mathrm{T}}(I - B^{-1}A)^{mr}\mathbf{e}| \leqslant 1$ and $\mathbf{e}_s^{\mathrm{T}}B^{-\sigma}A\mathbf{e} \geqslant 0$ are necessary for the $A(0)$-stability of the iterated RKN method.

Example 5.2. In the case of the $A(0)$-stable, third-order Radau based RKN method (2.8), we find for $B = A$ that $|\mathbf{e}_s^{\mathrm{T}}(I - B^{-1}A)^{mr}\mathbf{e}| = 0$ for all mr, but already for $\sigma = 2$ we have $\mathbf{e}_s^{\mathrm{T}}B^{-\sigma}A\mathbf{e} = \mathbf{e}_s^{\mathrm{T}}A^{1-\sigma}\mathbf{e} = -14$. Hence, according to theorem 5.2, we cannot have $A(0)$-stability. Figure 3 presents numerical plots for a few values of mr.

However, if we define B by (4.15), then the first condition is still satisfied because the spectral radius of $I - B^{-1}A$ vanishes and, hence, $(I - B^{-1}A)^{mr}$ vanishes for $mr \geqslant 2$ ($s \times s$ matrices M with only zero eigenvalues have the property that $M^n = \mathrm{O}$ for

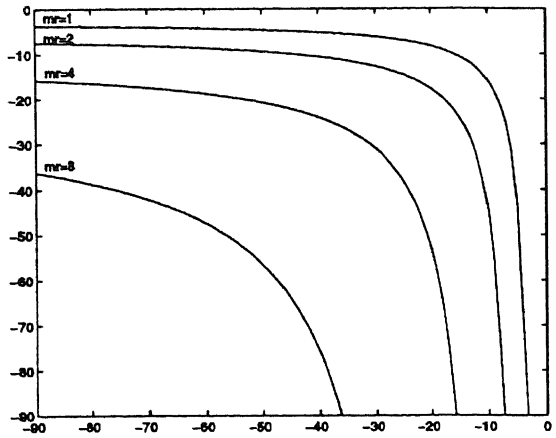


Figure 3. Stability boundaries in the (z_1, z_2)-plane for AF iteration applied to the RKN method generated by third-order Radau IIA method (2.8) with $B = A$.

$n \geqslant s$). Furthermore, it follows from (4.16) that $\mathbf{e}_s^{\mathrm{T}} B^{-\sigma} A\mathbf{e} = 2^{\sigma-1}$, so that the second necessary $A(0)$-stability condition of theorem 5.2 is also satisfied. Numerical plots for $\sigma = 2$ show $A(0)$-stability for all values of mr.

6. Concluding remarks

In this paper, we have analyzed an outer–inner iteration method based on modified Newton and approximate factorization for solving the implicit relations occurring in General Linear Methods (GLMs) for special second-order ODEs of the form (1.1). The implicit relations are characterized by a matrix A, the iteration method by a matrix B.

Convergence conditions can be expressed in terms of spectral properties of the matrices A and B. Table 2 summarizes the main convergence results for second-order equations as derived in the present paper and table 3 compares them with the A-convergence results for first-order equations derived in [3]. In these tables, A_{R3} indicates the Butcher matrix of the third-order Radau IIA method for first-order ODEs, and $\widetilde{A}$ and $\widetilde{B}$ refer to the matrices used in AF iteration for first-order ODEs.

The stability conditions for the AF iterated methods depend on the product mr of the number of outer and inner iterations. Easy to check conditions that are necessary for $A(0)$-stability have been derived for a family of Runge–Kutta–Nyström (RKN) methods. Tables 4 and 5 list the main results.

Table 2
Second-order ODEs. Cases of $A(0)$-convergence.

σ	$B = A$	$\rho(I - B^{-1}A) = 0$
2	$\mathrm{Re}(\lambda(A)) \geqslant 0$	$A = A_{\mathrm{R3}}^2$
3	$\lvert\arg(\lambda(A))\rvert \leqslant \pi/4$	$A = A_{\mathrm{R3}}^2$
$\geqslant 4$	$\lambda(A) \geqslant 0$	

Table 3
First-order ODEs. Cases of A-convergence.

σ	$\widetilde{B} = \widetilde{A}$	$\rho(I - \widetilde{B}^{-1}\widetilde{A}) = 0$
2	$\lambda(\widetilde{A}) \geqslant 0$	$\widetilde{A} = A_{R3}$

Table 4
Second-order ODEs. Cases of $A(0)$-stability.

σ	$B = A$	$\rho(I - B^{-1}A) = 0$
2	$A = A_{R3}^2$, $mr = \infty$	$A = A_{R3}^2$, $mr \geqslant 1$

Table 5
First-order ODEs. Cases of A-stability.

σ	$\widetilde{B} = \widetilde{A}$	$\rho(I - \widetilde{B}^{-1}\widetilde{A}) = 0$
2		$\widetilde{A} = A_{R3}$, $mr \geqslant 1$

References

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

[2] C.A. Coulson, *Waves, a Mathematical Account of the Common Types of Wave Motion* (Oliver and Boyd, Edinburgh, 1958).

[3] C. Eichler-Liebenow, P.J. van der Houwen and B.P. Sommeijer, Analysis of approximate factorization in iteration methods, Appl. Numer. Math. 28 (1998) 245–258.

[4] G.H. Golub and C.F. van Loan, *Matrix Computations* (North Oxford Academic, 1989).

[5] E. Hairer, Unconditionally stable methods for second-order differential equations, Numer. Math. 32 (1979) 373–379.

[6] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations*, Vol. II. *Stiff and Differential–Algebraic Problems* (Springer, Berlin, 1991).

[7] P.J. van der Houwen, B.P. Sommeijer and J. Kok, The iterative solution of fully implicit discretizations of 3-dimensional transport models, Appl. Numer. Math. 25 (1997) 243–256.

[8] D.W. Peaceman and H.H. Rachford Jr., The numerical solution of parabolic and elliptic differential equations, J. Soc. Indust. Appl. Math. 3 (1955) 28–41.

[9] L.F. Shampine, Implementation of implicit formulas for the solution of ODEs, SIAM J. Sci. Statist. Comput. 1 (1980) 103–118.

[10] P.W. Sharp, J.H. Fine and K. Burrage, Two-stage and three-stage diagonally implicit Runge–Kutta–Nyström methods of orders three and four, IMA J. Numer. Anal. 10 (1990) 489–504.

[11] C.B. Vreugdenhil, *Numerical Methods for Shallow-Water Flow* (Kluwer Academic, Dordrecht, 1994).

[12] A.G. Webster, *Partial Differential Equations of Mathematical Physics* (Dover, New York, 1933).