Journal of Computational and Applied Mathematics 231 (2009) 49–66

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

An iterated Radau method for time-dependent PDEs

S. Perez-Rodriguez [a,*], S. Gonzalez-Pinto [a], B.P. Sommeijer [b]

[a] *Dpto. de Analisis Matematico, Universidad de La Laguna, 38271 La Laguna-Tenerife, Spain*

[b] *CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands*

ARTICLE INFO

Article history:
Received 3 October 2008
Received in revised form 7 January 2009

MSC:
65M12
65M20

Keywords:
Advection-diffusion-reaction equations
Numerical integration
Single-Newton iteration
Approximate matrix factorization

ABSTRACT

This paper is concerned with the time integration of semi-discretized, multi-dimensional PDEs of advection-diffusion-reaction type. To cope with the stiffness of these ODEs, an implicit method has been selected, viz., the two-stage, third-order Radau IIA method. The main topic of this paper is the efficient solution of the resulting implicit relations. First, a modified Newton process has been transformed into an iteration process in which the 2 stages are decoupled and, moreover, can exploit the same LU-factorization of the iteration matrix. Next, we apply a so-called Approximate Matrix Factorization (AMF) technique to solve the linear systems in each Newton iteration. This AMF approach is very efficient since it reduces the 'multi-dimensional' system to a series of 'one-dimensional' systems. The total amount of linear algebra work involved is reduced enormously by this approach. The idea of applying AMF to two-dimensional problems is quite old and goes back to Peaceman and Rachford in the early fifties. The situation in three space dimensions is less favourable and will be analyzed here in more detail, both theoretically and experimentally. Furthermore, we analyze a variant in which the AMF-technique has been used to really solve ('until convergence') the underlying Radau IIA method so that we can rely on its excellent stability and accuracy characteristics. Finally, the method has been tested on several examples. Also, a comparison has been made with the existing codes VODPK and IMEXRKC, and the efficiency (CPU time versus accuracy) is shown to be at least competitive with the efficiency of these solvers.

1. Introduction

We are concerned with the numerical time integration of initial-value problems (IVP) for systems of ordinary differential equations (ODEs) of the form

$$y'(t) = f(t, y(t)), \qquad y(t_0) = y_0, \quad t_0 \le t \le t_{end}, \tag{1.1}$$

where $y, f \in \mathbb{R}^m$. Throughout the paper, these systems are assumed to be the result of applying a spatial discretization to a time-dependent partial differential equation (PDE). Hence, we follow the Method of Lines (MoL) approach.

The literature on the time integration of the resulting system of ODEs is overwhelming, which is caused by the widely varying nature of the underlying PDEs. Numerical processes that behave efficiently for one particular class of PDEs are not necessarily a good choice for other classes. For example, methods suitable for hyperbolic problems are often of a completely different concept compared with methods for parabolic problems. Moreover, many 'industrial problems' are so specific that they justify an ad hoc approach and are best solved by a method that is tuned to their idiosyncrasies. Nevertheless, one can

* Corresponding author. Tel.: +34 922319071; fax: +34 922319230.
E-mail addresses: sperezr@ull.es (S. Perez-Rodriguez), spinto@ull.es (S. Gonzalez-Pinto), ben.sommeijer@cwi.nl (B.P. Sommeijer).

0377-0427/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.01.020

try to design algorithms for problem classes as wide as possible. The major aim of this paper is to come up with such an algorithm.

In designing such a time integration method one has to identify certain common characteristics of the underlying PDE classes that the numerical method is capable to cope with. For example, a typical property of systems (1.1) is that they possess stiffness; that is, the eigenvalues of the Jacobian matrix $\partial f/\partial y$ differ largely in magnitude. The stiffness can be substantial if the PDE has to be semi-discretized on a spatial grid with high (local) resolution to meet certain accuracy conditions. Another aspect, related to the concept of stiffness, is that – apart from advection and diffusion operators – often stiff reaction terms are involved. Such a situation is e.g. exemplified in chemical reactions which typically have widely varying time scales.

Another complicating factor for dealing with stiffness is that the eigenvalues of $\partial f/\partial y$ can be situated anywhere in the negative half plane. For example, diffusion-reaction terms often give rise to negative real eigenvalues, but the discretization of advection terms usually leads to eigenvalues possessing a substantial imaginary part. The above considerations lead us to aim for a numerical time integrator that is capable to treat ODEs independent of the position of the eigenvalues in the left half plane. In other words, we will require the method to be A-stable [1]. As a consequence of this choice we shall exclude all explicit methods. Confining ourselves to the class of implicit methods, there is still a considerable choice: a well-known class of methods is given by the BDF methods; indeed the popular and widely used codes VODE [2] and VODPK[1] [4,5] are based on this class. However, since the pioneering work of Dahlquist [6] we know that the order of A-stable methods of this type is necessarily limited to 2. On the other hand, the amount of implicitness of these methods is minimal which explains their popularity.

An alternative, to circumvent the order-2 barrier with respect to A-stability, is offered by the class of implicit Runge–Kutta (IRK) methods. For example, the code RADAU5 in [1] is based on this concept and is a robust and accurate stiff ODE solver. The amount of implicitness, however, is larger than for VODE, due to the IRK-nature. Based on the above considerations, we have decided to select a member from the IRK-family as our starting point to build a robust solver. To be more specific, we have chosen the 2-stage Radau IIA scheme. This method combines excellent stability properties (i.e., even the stronger concept of L-stability, see [1]) with order of accuracy equal to 3, which we think is an appropriate choice in a PDE context.

No matter which implicit method has been selected, we are always faced with solving implicit relations to obtain the numerical approximation in the new step point. In fact, solving these systems is the determining factor for the success of a PDE-solver. This is particularly true in case of *multi*-dimensional PDEs where a straightforward approach of the linear algebra involved may easily lead to excessive costs. To further elaborate this, let us consider the 2-stage Radau IIA method. Applying this method, we encounter 2 main difficulties:

(i) apart from computing a new step point approximation, the scheme requires to solve for a (coupled) intermediate approximation; this requirement doubles the dimension of the algebraic systems to be solved in each step, and

(ii) the sparsity patterns in the matrices involved in the Newton process require – especially for three-dimensional PDEs – a special treatment since standard LU-decompositions are not feasible in such cases.

To cope with the first difficulty, Butcher proposed already in 1976 [7] a similarity transformation to reduce the dimension of the implicit system to solve. The idea of only solving systems of dimension m has been exploited in many papers [8–11]. Techniques to 'decouple' the stages are based on properties of the A-matrix in the RK scheme; also the code RADAU5 is based on this principle. The approach to be discussed in the present paper follows the same idea: the classical Newton iteration for the full implicit relation is replaced by a much simpler iteration in which the stages are decoupled and hence only systems of dimension m have to be solved. The definition of this iteration, as well as an analysis of its convergence behaviour will be described in Section 2. In passing we remark that also SDIRK methods efficiently reduce the implicitness to a dimension equal to that of the ODE system. In this context, however, we have to mention a phenomenon called 'order reduction'. Almost all A-stable Runge–Kutta methods, when applied to arbitrary stiff ODEs, might show an actual order of convergence that is smaller than the classicial order. This order reduction is largely determined by the so-called stage order. Typically, SDIRK methods have stage order equal to one, resulting in an actual order equal to one or two, see e.g.[12, Th. 3.3] and [13, Th. 3.4] for conditions and further details. For semi-linear problems it has been shown that the actual order of convergence of s-stage Radau IIA methods is given by $\min\{2s-1, s+1\}$. For the 2-stage Radau IIA method that will be used in this paper, this yields third-order convergence. A clarifying discussion on order reduction suffered by Runge–Kutta methods applied in a PDE context can be found in [14, Ch.II.2].

For the second difficulty, i.e., the structure of the Jacobian matrices originating from a multi-dimensional PDE, we use a so-called Approximate Matrix Factorization (AMF) approach. Also this idea is already quite old. In fact, the celebrated paper of Peaceman and Rachford from the early fifties was one of the first based on this principle. However, so far a successful application of AMF was usually restricted to two-dimensional problems. In Section 3 we will discuss an extension suitable for three spatial dimensions. This idea originates from the overview paper [15], but in that paper it was only suggested as a possible treatment. As far as we know this idea has not yet been tested in real life three-dimensional applications. Hence, the above techniques are not novel; what is new – and that is the main contribution of this paper – is the combination of both ingredients into one overall approach to tackle multi-dimensional PDEs by keeping the costs to deal with the implicitness to a manageable level.

[1] VODPK is based on VODE, extended with the Krylov solver GMRES [3] allowing for a user-supplied preconditioner to accelerate convergence of the iteration process to solve the systems.

Next, the performance of the resulting algorithm is demonstrated on several test problems. We start with a linear model problem in Section 4 to study the basic properties of the combined method. Then, in Section 5, the method is applied to a realistic, nonlinear problem and will be compared with existing solvers, such as VODPK and IMEX RKC. Finally, some conclusions will be formulated in Section 6.

2. Single-Newton iteration

Applying a fully implicit s-stage RK method to the ODE system (1.1) leads to

$$\begin{aligned} Y_n &= e \otimes y_n + \tau (A \otimes I_m) F(et_n + c\tau, Y_n), \\ y_{n+1} &= y_n + \tau (b^{\mathrm{T}} \otimes I_m) F(et_n + c\tau, Y_n), \end{aligned} \tag{2.1}$$

where the RK method is characterized by the matrix A and the vector b (both of dimension s), Y_n is the so-called stage vector, containing the s approximations $Y_{n,i} \approx y(t_n + c_i\tau)$, $i = 1, \ldots, s$ with τ being the step size and c_i are the elements of the collocation vector $c = Ae$. Furthermore, $F(et_n + c\tau, Y_n)$ contains the f-evaluations at the collocation points, i.e., $F(et_n + c\tau, Y_n) = (f(t_n + c_1\tau, Y_{n,1})^{\mathrm{T}}, \ldots, f(t_n + c_s\tau, Y_{n,s})^{\mathrm{T}})^{\mathrm{T}}$, I_m is the m-dimensional identity matrix, e is the s-dimensional vector with unit entries, and $\otimes$ denotes the Kronecker product. The quantity y_{n+1} is an approximation to the solution $y(t)$ at $t = t_{n+1} = t_n + \tau$.

The usual approach in a stiff context is to solve the stage vector Y_n from (2.1) by means of a modified Newton iteration

$$\begin{aligned} &[I_{ms} - \tau A \otimes J]\Delta^k = D^{k-1}, \\ &Y_n^k = Y_n^{k-1} + \Delta^k, \quad k = 1, 2, \ldots, \end{aligned} \tag{2.2}$$

where the residual D^{k-1} is defined by

$$D^{k-1} = e \otimes y_n - Y_n^{k-1} + \tau (A \otimes I_m) F(et_n + c\tau, Y_n^{k-1}), \tag{2.3}$$

and J is an approximation to the Jacobian $\frac{\partial f}{\partial y}(t_n, y_n)$. The iteration is started with Y_n^0, provided by some predictor formula. To simplify the presentation, here and henceforth we omit the dependence on n of any residual D^{k-1}.

In each iteration of (2.2) a linear system of dimension $s \cdot m$ has to be solved. As proposed by Butcher [7], a similarity transformation can be used to reduce the dimension. Unfortunately, for the s-stage implicit Runge–Kutta Radau IIA methods ($s \geq 2$), which we take as starting point, the A-matrix has pairs of conjugate complex eigenvalues. As a consequence, the Butcher-approach leads to solving (block) systems of dimension $2m$, or – alternatively – change to complex arithmetic.

Another approach, which has been considered in several papers [8–11], is to replace the matrix A in the left-hand side of (2.2) by a 'more convenient' matrix T. By 'more convenient' we mean that the matrix T has a structure by which the stages are decoupled (so that only systems of dimension m have to be solved, independent of the number of stages s) and, moreover, T has a one-point spectrum, so that only one LU-decomposition of an $m \times m$ matrix is required.

In the papers mentioned above, the matrix T is determined on the basis of a linear analysis. Here, we follow a similar approach, that is we apply the iteration scheme (2.2) with A (in the left-hand side) replaced by T to the scalar linear equation $y' = \lambda y$, with $\mathbb{R}e\ \lambda \leq 0$, and find that the iteration error $\varepsilon^k := Y_n^k - Y_n$ satisfies the recursion

$$\varepsilon^k = M(z)\varepsilon^{k-1}, \qquad M(z) = z(I_s - zT)^{-1}(A - T), \quad z = \tau\lambda. \tag{2.4}$$

Clearly, for convergence we need that the spectral radius ρ of the iteration matrix satisfies $\rho(M(z)) < 1$. If this iteration process converges, then it converges to the solution of the underlying Radau IIA method and we can take full profit of the accurate and stable behaviour of this corrector.

To determine a suitable T-matrix, we follow an approach as suggested in [10]. In that paper, requirements on a suitable rate of convergence are combined with adequate linear stability properties, both for $|z| \to \infty$, i.e. the focus is on extremely stiff components. These two conditions respectively lead to

$$\rho(M(\infty)) = \rho(I_s - T^{-1}A) = 0 \tag{2.5}$$

and

$$b^{\mathrm{T}}A^{-2}(A - T) = 0^{\mathrm{T}}. \tag{2.6}$$

Here, we remark that (2.6) implies that $R_k(\infty) = R(\infty)$ for all $k \geq 1$, where $R_k(z)$ is the stability function for the advancing solution of the method obtained after k iterations, namely y_{n+1}^k, starting with the predictor $Y_n^0 = e \otimes y_n$, and $R(z)$ is the stability function of the underlying IRK method, henceforth called the corrector. Since we will use an L-stable Radau IIA method as corrector, which is stiffly accurate ($y_{n+1}^k = Y_{n,s}^k$), we have – after an arbitrary number of $k \geq 1$ iterations – that the resulting stability function satisfies $R_k(\infty) = 0$. Another result, which we will use in the numerical Sections 4 and 5, concerns the order of accuracy of the overall method. It is well-known that the order of accuracy is increased by one in each Single-Newton iteration until the order of the underlying corrector has been reached. It must be noted that this fact is

independent of the approximation J taken in the iterative scheme to replace $\partial f/\partial y(t_n, y_n)$ as long as $J - \partial f/\partial y(t_n, y_n) = \mathcal{O}(1)$. Hence after, say q iterations, the order p^* of the advancing solution y_{n+1}^q equals

$$p^* = \min(\ell + q, p), \tag{2.7}$$

where p is the order of the corrector and ℓ is the order of the prediction Y_n^0, i.e., $Y_n^0 - Y_n = \mathcal{O}(\tau^{\ell+1})$. For additional properties of this iteration process we refer to [10].

We will now derive the matrix T. Recalling that we require T to have a one-point spectrum, this matrix can be written as

$$T = \gamma S(I_s - L)^{-1} S^{-1} \tag{2.8}$$

where L is a strictly lower triangular matrix, S is nonsingular, and γ is the multiple eigenvalue, which needs to be positive. Replacing the matrix A in (2.2) by T and using its decomposition (2.8), we arrive at what we will call the single-Newton iteration process [10],

$$\begin{aligned} &[I_{ms} - \gamma\tau(I_s \otimes J)]E^k = ((I_s - L)S^{-1} \otimes I_m)D^{k-1} + (L \otimes I_m)E^k, \\ &Y_n^k = Y_n^{k-1} + (S \otimes I_m)E^k, \quad k = 1, 2, \ldots. \end{aligned} \tag{2.9}$$

Since L is strictly lower triangular, the s components $E_1^k, \ldots, E_s^k$ can be solved one after another and hence, only systems of dimension m are involved. We remark that (2.9) can be considered as a special case of the class of iteration methods considered in [9].

2.1. Determining the matrix T for the 2-stage Radau IIA corrector

As motivated in the introduction, an appropriate choice, in a PDE context, for the corrector is the third-order, 2-stage Radau IIA method, defined by

$$A = \begin{pmatrix} \frac{5}{12} & -\frac{1}{12} \\ \frac{3}{4} & \frac{1}{4} \end{pmatrix}, \qquad b^{\mathrm{T}} = \left(\frac{3}{4}, \frac{1}{4}\right).$$

Since this method is stiffly-accurate, i.e., $b^{\mathrm{T}}A^{-1} = (0, 1)$, we have that y_{n+1} equals the second stage vector component $Y_{n,2}$. To determine a matrix T that satisfies the conditions (2.5) and (2.6), we define the matrix P by

$$P = I_2 - A^{-1}T.$$

Clearly, condition (2.5) is equivalent to the requirement that both eigenvalues of P vanish. Furthermore, condition (2.6) now reads $b^{\mathrm{T}}A^{-1}P = 0^{\mathrm{T}}$, which, for this corrector, leads to the requirement that the second row of P is the zero vector. Hence, P is of the form

$$P = \begin{pmatrix} 0 & b \\ 0 & 0 \end{pmatrix}.$$

The matrix T has a double eigenvalue γ iff

$$\det T = \gamma^2 \quad \text{and} \quad \operatorname{trace} T = 2\gamma.$$

Since $T = A(I_2 - P)$ we have $\det T = \det A$ and then γ is determined by

$$\gamma = \sqrt{\det A} = \frac{1}{6}\sqrt{6}, \tag{2.10}$$

which is positive indeed, as required. Using $\operatorname{trace} T = 2\gamma = \frac{8-9b}{12}$, we have uniquely determined the matrices P and T as

$$P = \begin{pmatrix} 0 & \frac{8-4\sqrt{6}}{9} \\ 0 & 0 \end{pmatrix}, \qquad T = \begin{pmatrix} \frac{5}{12} & \frac{5\sqrt{6}}{27} - \frac{49}{108} \\ \frac{3}{4} & \frac{\sqrt{6}}{3} - \frac{5}{12} \end{pmatrix}. \tag{2.11}$$

Given the matrix T, we finally have to determine its decomposition (2.8). Setting $R = (I_2 - L)^{-1}$, the matrices S and R have the form

$$S = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}, \qquad R = \begin{pmatrix} 1 & 0 \\ x & 1 \end{pmatrix},$$

and they have to satisfy the equation $TS = \gamma SR$. In solving this system we are left with three free parameters $\{x_1, x_2 \neq 0, x_3\}$. However, it is not possible to exploit this freedom to obtain better damping properties of the iteration scheme. Therefore, we

will use the approach used in [10] where the transformation matrix S has been chosen upper triangular with unit diagonal entries; this facilitates the implementation and reduces the computational costs. This choice leads to

$$S = \begin{pmatrix} 1 & \frac{5-2\sqrt{6}}{9} \\ 0 & 1 \end{pmatrix}, \qquad L = I_2 - R^{-1} = \begin{pmatrix} 0 & 0 \\ \frac{3\sqrt{6}}{4} & 0 \end{pmatrix}.$$

The matrix $(I_2 - L)S^{-1}$, also needed in the single-Newton process (2.9), is given by

$$(I_2 - L)S^{-1} = \begin{pmatrix} 1 & -\frac{5-2\sqrt{6}}{9} \\ -\frac{3\sqrt{6}}{4} & \frac{5\sqrt{6}}{12} \end{pmatrix}.$$

We conclude this section by mentioning that one eigenvalue of the iteration matrix $M(z)$ identically vanishes. For the other eigenvalue an analytical expression can be derived:

$$\rho(M(z)) = |\text{trace } M(z)| = \left| \frac{2(2-\sqrt{6})z}{(\sqrt{6}-z)^2} \right|,$$

for which we have the following suprema along the negative real axis and the imaginary axis

$$\max_{z\le 0}\{\rho(M(z))\} = \frac{1}{2} - \frac{\sqrt{6}}{6} \approx 0.09175, \qquad \max_{y\in\mathbb{R}}\{\rho(M(iy))\} = 1 - \frac{2\sqrt{6}}{6} \approx 0.18350.$$

3. Approximate matrix factorization

The single-Newton iteration process (2.9) requires, in each iteration, the solution of the two m-dimensional linear systems

$$\begin{cases} (I_m - \gamma\tau J)E_1^k = \tilde{D}_1^{k-1} \\ (I_m - \gamma\tau J)E_2^k = \tilde{D}_2^{k-1} + L_{21}E_1^k \end{cases} \tag{3.1}$$

where we have put

$$E^k = \begin{pmatrix} E_1^k \\ E_2^k \end{pmatrix}, \qquad \tilde{D}^{k-1} = \begin{pmatrix} \tilde{D}_1^{k-1} \\ \tilde{D}_2^{k-1} \end{pmatrix} := ((I_2 - L)S^{-1} \otimes I_m)D^{k-1}. \tag{3.2}$$

Notice that the coupling in the two systems in (3.1) is one-sided, which implies that first E_1^k can be computed and subsequently E_2^k, using E_1^k in the right-hand side. In the current application of *multi*-dimensional PDEs, the direct solution of these linear systems is time consuming, due to the structure of the Jacobian. A possible remedy to reduce the computational costs is to use a so-called Approximate Matrix Factorization (AMF) technique. To that end, the Jacobian matrix J is written as $J = \sum_{i=1}^{d} J_i$. Then the matrices $I_m - \gamma\tau J = I_m - \gamma\tau(J_1 + \cdots + J_d)$ in (3.1) are replaced by the factored matrix Π, defined as

$$\Pi := \prod_{i=1}^{d}(I_m - \gamma\tau J_i). \tag{3.3}$$

In this paper, d will be chosen equal to the number of spatial dimensions of the underlying PDE, and J_i corresponds to the discretization of the differential operators in the i-th spatial direction. Solving the resulting linear systems is much cheaper because the factored matrix Π results in the successive solution of d systems with a *banded* coefficient matrix. Typically, the matrices have a band width in the range 3–5, depending on the discretization stencils that have been used (e.g., symmetric second-order for diffusion terms, third-order upwind biased for advection terms, etc.). Solving such systems is cheap since the complexity involved is only *linear* in the dimension. Now, we can proceed in two different directions.

3.1. Factorized iteration to solve the linear systems

The first approach is to use the AMF-techniques in an *iterative* way to solve the linear systems in (3.1) until 'convergence'. This is in the spirit of the analysis of the single-Newton iteration. Indeed, the use of the expression $(I_s - zT)^{-1}$ in the derivation of the iteration matrix $M(z)$ (cf. (2.4)) assumes that the linear system is exactly solved. The convergence behaviour of this AMF-iteration has been analyzed in [15]; see also [16] with a similar analysis in a slightly different context. As it turns out, a successful application of the AMF approach critically depends on the number of spatial dimensions involved.

Writing each of the linear systems in (3.1) in the form $(I_m - \gamma\tau J)x = b$, the convergence of the AMF-iteration process

$$\Pi(x^j - x^{j-1}) = b - (I_m - \gamma\tau J)x^{j-1}, \quad j = 1, 2, \ldots, \tag{3.4}$$

corresponding to the linear model problem $y' = Jy = (J_1 + \cdots + J_d)y$ is governed by the iteration matrix Z given by

$$Z = I_m - \Pi^{-1}(I_m - \gamma\tau J). \tag{3.5}$$

Assuming that all the Jacobian matrices J_i $(i = 1, \ldots, d)$ have the same set of eigenvectors, then the eigenvalues of Z are given by

$$\lambda(Z) = 1 - (1 - \gamma z)\prod_{i=1}^{d}(1 - \gamma z_i)^{-1}, \tag{3.6}$$

where z_i runs through the eigenvalues of τJ_i and $z = \sum_{i=1}^{d} z_i$.

The process (3.4) is called A(α)-convergent [15] if $\lambda(Z)$ is within the unit circle for all $z_i \in \mathcal{W}(\alpha)$ with

$$\mathcal{W}(\alpha) := \{w \in \mathbb{C} : w = 0 \quad \text{or } |\arg(-w)| < \alpha\}.$$

Now, we have the following

Theorem 1 ([16,15]). *For the convergence of the AMF-iteration process (3.4) we have for $d \geq 2$*

$$|\lambda(Z)| < 1 \quad \text{for all } z_i \in \mathcal{W}(\alpha) \Longleftrightarrow \alpha \leq \frac{1}{d-1}\cdot\frac{\pi}{2}. \quad \square$$

For PDEs in *two* spatial dimensions this result is excellent, since $d = 2$ yields A($\pi/2$)-convergence, hence unconditional convergence as long as the eigenvalues of J_1 and J_2 are in the left half-plane. On the other hand, we encounter a serious limitation for PDEs in *three* dimensions, since then we only have A($\pi/4$)-convergence. This implies that for advection dominated 3D PDEs, we will encounter convergence problems. In passing we remark that, if J has only real negative eigenvalues (corresponding to diffusion-reaction type PDEs without advection terms), the process will converge, independent of the number of dimensions d.

In [15] a remedy has been suggested to circumvent this restrictive condition on α in case of 3D PDEs. The basic idea is to replace the factorization $\prod_{i=1}^{3}(I_m - \gamma\tau J_i)$ by two successive factorizations in each of which only two matrices are involved. Writing

$$J = J_1 + J^*, \quad \text{with } J^* = J_2 + J_3, \tag{3.7}$$

and recursively applying the $d = 2$-application of the AMF-iteration with these matrices, we arrive at

$$\begin{aligned} &(I_m - \gamma\tau J_1)\tilde{\Delta}^j = b - (I_m - \gamma\tau J)x^{j-1}, \\ &(I_m - \gamma\tau J^*)\Delta^j = \tilde{\Delta}^j, \qquad x^j = x^{j-1} + \Delta^j, \quad j = 1, 2, \ldots. \end{aligned} \tag{3.8}$$

The matrix J_1 has a simple band structure, but J^* has not. Therefore, the system involving J^* is iteratively solved by a (nested) AMF-iteration. Since both processes are based on a factorization with $d = 2$, they will converge unconditionally. Now, the inner AMF-iteration is obtained by replacing $I_m - \gamma\tau J^*$ by $(I_m - \gamma\tau J_2)(I_m - \gamma\tau J_3)$ which results in

$$(I_m - \gamma\tau J_2)(I_m - \gamma\tau J_3)(\Delta^{j,i} - \Delta^{j,i-1}) = \tilde{\Delta}^j - (I_m - \gamma\tau J^*)\Delta^{j,i-1} \tag{3.9}$$

for $i = 1, 2, \ldots, r$, and the vector x^j is updated by the last result from this inner iteration, i.e., $x^j = x^{j-1} + \Delta^{j,r}$. We remark that the inner iteration should be continued until 'convergence', hence r should be sufficiently large. A plausible starting value for the iteration (3.9) is given by $\Delta^{j,0} = \tilde{\Delta}^j$, as has been suggested in [15]. For the approach described in the next subsection, however, there is theoretical and numerical evidence that $\Delta^{j,0} = 0$ is a better choice to start the iteration.

In applying the above (nested) AMF-iteration process to really solve the linear systems in (3.1), the overall behaviour of the combined single-Newton/AMF process is merely governed by the convergence behaviour of the single-Newton process, which has been analyzed in Section 2.

3.2. Mixed single-Newton and AMF-iteration

Next, we will discuss an approach in which both iteration processes are mixed up. By this we mean that the linear systems (3.1) that occur in each single-Newton iteration are only *approximately* solved by replacing the matrix $I_m - \gamma\tau J$ by the matrix Π defined in (3.3). Then, after successively solving the d bandsystems, we continue with the next single-Newton iteration. Or, saying it differently, only one AMF-iteration of the form (3.4) is applied. This approach requires, of course, much less bandsolves than the approach discussed in the preceding subsection. The convergence analysis, however, does not directly follow from the results given in [15,16] and needs some amendment. Starting from (3.1) and applying the AMF-technique, this mixed approach reads (see also (2.9))

$$\left.\begin{aligned} &\Pi E_1^k = \tilde{D}_1^{k-1}, \Pi E_2^k = \tilde{D}_2^{k-1} + L_{21}E_1^k, \\ &Y_n^k = Y_n^{k-1} + (S \otimes I_m)E^k, \quad k = 1, 2, \ldots, \end{aligned}\right\} \tag{3.10}$$

with Π and $\tilde{D}_j^{k-1}$ given by (3.3) and (3.2), respectively. A natural initial guess is given by $Y_n^0 = e \otimes y_n$, but other choices are possible, e.g. predictions of higher order (i.e., with $\ell > 0$ in (2.7)).

If the mixed iteration process (3.10) is applied to $y' = Jy = (J_1 + \cdots + J_d)y$, we find that the iteration error

$$\varepsilon^k := (S^{-1} \otimes I_m)(Y_n^k - Y_n) \tag{3.11}$$

satisfies the recursion $\varepsilon^k = Z^* \varepsilon^{k-1}$, $k = 1, 2, \ldots$, where

$$\begin{aligned} Z^* &= I_{2m} - (I_2 \otimes \Pi^{-1}) \left[I_{2m} + L \otimes (\Pi^{-1} - I_m) \right] (I_{2m} - \tau \tilde{A} \otimes J), \\ \tilde{A} &:= S^{-1} A S, \end{aligned} \tag{3.12}$$

and we have taken into account that $L^2 = 0$.

Similarly as in Section 3.1, it is assumed that all the J_i have the same set of eigenvectors, that z_i runs through the spectrum of τJ_i and $z = \sum_{i=1}^d z_i$. Then, the eigenvalues of Z^* are those of the 2-dimensional matrix M^*

$$M^* = I_2 - x^{-1} \left[I_2 + (x^{-1} - 1)L \right] (I_2 - z\tilde{A}) \tag{3.13}$$

where $x = \prod_{j=1}^d (1 - \gamma z_j)$.

Next, we formulate convergence results for the cases $d = 2$ and $d = 3$.

Result 1. The iteration process (3.10) with $d = 2$ is convergent for $z_1, z_2 \in \mathcal{W}(\alpha)$ with $\alpha \approx 87.9°$.

Derivation. A straightforward calculation yields that the eigenvalues μ of the matrix M^* are determined by

$$\begin{aligned} &\mu^2 - a_1 \mu + a_0 = 0, \\ &a_0 = \left(6 + 6x^2 - 2\sqrt{6}z + z^2 + 2x(-6 + \sqrt{6}z) \right) (x^{-2}/6), \\ &a_1 = \left(6x^2 - (-2 + \sqrt{6})z + x(-6 + \sqrt{6}z) \right) (x^{-2}/3). \end{aligned} \tag{3.14}$$

Since z_1 and z_2 may vary independently in the wedge $\mathcal{W}(\alpha)$, we examine the cases $\{z_1 = \rho_1 e^{i\alpha}, z_2 = \rho_2 e^{i\alpha}\}$ and $\{z_1 = \rho_1 e^{i\alpha}, z_2 = \rho_2 e^{-i\alpha}\}$ and determine numerically the largest α such that $\rho(M^*) < 1$ for (many values of) ρ_1 and $\rho_2 \in (0, \infty)$. This computation yields $\alpha \approx 87.9°$. We remark that the largest values for the spectral radius in any interval $0 \le \rho_1, \rho_2 \le \rho$, $(\rho > 0)$ were found when $\rho_1 = \rho_2 \in [0, \rho]$. □

To obtain a convergence result for the three-dimensional case we followed the same approach using $d = 3$. That is, all possible combinations of z_i-values lying on the upper and lower boundary of the wedge and at mutually different distances from the origin have been examined. A numerical search for the largest aperture of the wedge, still resulting in convergence, leads to the following result.

Result 2. The iteration process (3.10) with $d = 3$ is convergent for $z_1, z_2, z_3 \in \mathcal{W}(\alpha)$ with $\alpha \approx 44.7°$.

Remark 1. It is interesting to compare the convergence properties of the mixed iteration process with those of the approach described in Section 3.1. To this aim, by comparing the Results 1 and 2 with Theorem 1, we conclude that the angle α reduces from 90° to 87.9° in case $d = 2$, and from 45° to 44.7° for $d = 3$. This marginal reduction of the convergence region is amply compensated by the enormous gain in computational work.

Remark 2. From (3.14) we obtain for $\mathbb{R}e\, z_j \to -\infty$ $(j = 1, \ldots, d)$, that $a_0 \to 1$ and $a_1 \to 2$. Then $\rho(M^*)$ is only slightly smaller than 1, indicating that we may expect slow convergence for extremely stiff components.

So far, we have considered the convergence of the single-Newton process (2.9) combined with AMF. One may wonder whether better convergence results are obtained if we apply the AMF-technique directly to the modified Newton process (2.2). Hence, when the iteration matrix $(I_{2m} - \tau A \otimes J)$ in (2.2) is replaced by $\prod_{j=1}^d (I_{2m} - \tau A \otimes J_j)$. As we will show, this approach leads to a wedge with smaller aperture. This negative result is due to the nonzero imaginary parts in the eigenvalues of the A-matrix of the 2-stage Radau IIA method. For this method, the eigenvalues are $\eta(A) = (2 \pm i\sqrt{2})/6 = \frac{\sqrt{6}}{6} e^{\pm i\alpha_R}$, with $\alpha_R = \arctan(\sqrt{2}/2) \approx 0.615$ $(\approx 35.3°)$. The new iteration can be written as

$$\prod_{j=1}^d [I_{2m} - \tau A \otimes J_j] \Delta^k = D^{k-1}, \qquad Y_n^k = Y_n^{k-1} + \Delta^k, \quad k = 1, 2, \ldots. \tag{3.15}$$

For linear problems $y' = Jy, J = \sum_{j=1}^d J_j$, the error of the iterates satisfies

$$\begin{aligned} &Y_n^k - Y_n = W^* (Y_n^{k-1} - Y_n) \\ &W^* = I_{2m} - \left(\prod_{j=1}^d [I_{2m} - \tau A \otimes J_j] \right)^{-1} (I_{2m} - \tau A \otimes J). \end{aligned} \tag{3.16}$$

Again, we assume that the J_j matrices share the same set of eigenvectors. Then, the eigenvalues of W^* are given by

$$\lambda(W^*) = 1 - \left(\prod_{j=1}^{d}(1 - z_j\eta(A))^{-1}(1 - z\eta(A))\right) \qquad \eta(A) = (2 \pm \mathrm{i}\sqrt{2})/6 \tag{3.17}$$

where the z_j and z have the same meaning as before. Now we can formulate the following theorem.

Theorem 2. *For the convergence of the iteration process (3.15) and the two-dimensional case ($d = 2$) we have*

$$|\lambda(W^*)| < 1 \quad \text{for all } z_1, z_2 \in \mathcal{W}(\alpha) \quad \Longleftrightarrow \quad \alpha \leq \pi/2 - \alpha_R \simeq 54.7^\circ.$$

Proof. Writing $\eta = \eta(A)$, it readily follows from (3.17) with $d = 2$ that

$$\lambda(W^*) = \frac{\eta^2 z_1 z_2}{(1 - \eta z_1)(1 - \eta z_2)} = \frac{\eta z_1}{(1 - \eta z_1)} \cdot \frac{\eta z_2}{(1 - \eta z_2)}.$$

Hence, $|\lambda(W^*)| < 1$, $\forall\ z_1, z_2 \in \mathcal{W}(\alpha)$ iff $|\eta\zeta(1 - \eta\zeta)^{-1}| < 1, \forall\, \zeta \in \mathcal{W}(\alpha)$. The latter expression is equivalent to $\alpha = \pi/2 - \alpha_R$. □

3.3. Stability analysis

As mentioned in Remark 2, the convergence of the mixed iteration process can be rather slow, especially for stiff eigenvalues close to the boundary of the wedge. Therefore, in computational practice we will not continue the iteration until the true Radau IIA solution has been reached. Starting with the prediction

$$Y_n^0 = e \otimes y_n, \tag{3.18}$$

we see from (2.7) that after $q \geq 3$ outer iterations, an advancing solution $y_{n+1} = Y_{n,2}^q$, of order $p = 3$ is obtained. For this reason, we will focus on applications with $q = 3$ or $q = 4$. In case of a 3D problem we will also employ the inner iteration process (nested AMF) as described in Section 3.1. Again, we are mainly interested in a small number of inner iterations r. It should be remarked that with $q = 3$ outer iterations, third-order accuracy is obtained (independent of the number of inner iterations), however the principal local error term will differ from the corresponding term of the Radau IIA corrector. With $q = 4$, however, the principal local error term coincides with that of the corrector.

Stopping the iteration process before convergence has been reached, implies that we cannot simply rely on the stability properties of the underlying Radau IIA corrector. Therefore, it is of interest to study the stability properties of the final approximation y_{n+1} obtained after a modest number of iterations.

Definition 1. A one-step method $y_{n+1} = \phi(t_n, y_n, \tau)$ is said to be A(α)-stable for the d-dimensional case, if its stability function $R(z_1, \ldots, z_d)$ satisfies $|R(z_1, \ldots, z_d)| \leq 1$, whenever z_j are in the closure of $\mathcal{W}(\alpha)$ for $j = 1, 2, \ldots, d$. In addition, if $\alpha = \pi/2$ the method is said to be A-stable.

Result 3. For the two-dimensional case ($d = 2$), the method $y_{n+1}^q = Y_{n,2}^q$ obtained from the mixed iteration process (3.10) with q iterations and with predictor (3.18), is A-stable for $q = 1, 2, 3, 4$.

Derivation. Applying the mixed iteration (3.10) to the test problem

$$y' = \left(\sum_{j=1}^{d}\lambda_j\right)y, \qquad z_j = \tau\lambda_j\ (j = 1, \ldots, d), \qquad z = \sum_{j=1}^{d} z_j, \tag{3.19}$$

it follows from (3.11) and (3.13) that

$$Y_n^k - Y_n = SM^*S^{-1}(Y_n^{k-1} - Y_n) = S(M^*)^kS^{-1}(Y_n^0 - Y_n). \tag{3.20}$$

From (2.1), the stage vector Y_n of the 2-stage Radau IIA method is seen to satisfy

$$Y_n = ey_n + zAY_n, \qquad y_{n+1} = e_2^{\mathrm{T}}Y_n, \quad e_2^{\mathrm{T}} = (0, 1).$$

Solving for Y_n and inserting the result into (3.20) leads to

$$Y_n^q = \left((I_2 - zA)^{-1}e + S(M^*)^qS^{-1}(I_2 - (I_2 - zA)^{-1})e\right)y_n.$$

Taking into account that the stability function of the advancing solution corresponding to q outer iterations is obtained by setting $y_{n+1}^q = e_2^{\mathrm{T}}Y_n^q \equiv R_q(z_1, \ldots, z_d)y_n$, it follows that

$$\begin{aligned} R_q(z_1, \ldots, z_d) &= R(z) + e_2^{\mathrm{T}}S(M^*)^qS^{-1}\left(I_2 - (I_2 - zA)^{-1}\right)e \\ &= R(z) + e_2^{\mathrm{T}}S(M^*)^q\left(I_2 - (I_2 - z\tilde{A})^{-1}\right)S^{-1}e, \end{aligned} \tag{3.21}$$

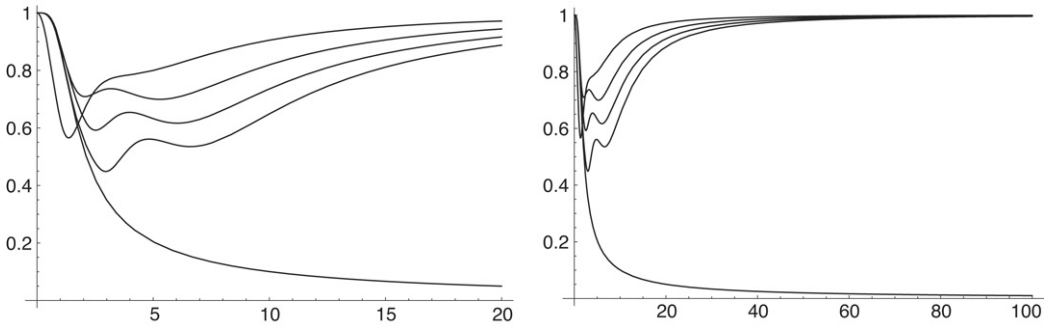

Fig. 3.1. Graphs of $|R(2t\,\mathrm{i})|$ and $|R_q(t\,\mathrm{i}, t\,\mathrm{i})|$ (vertical axis) for $q = 1, 2, 3, 4$ and $t \geq 0$ (horizontal axis). The left panel shows the situation in the neighborhood of the origin, whereas the right panel illustrates the behaviour on the larger interval $0 \leq t \leq 100$. The situation near the origin serves to distinguish between the curves for the various q-values: the larger q, the closer $|R|$ and $|R_q|$.

where $R(z) = e_2^{\mathrm{T}}(I - zA)^{-1}e = (1 + z/3)/(1 - 2z/3 + z^2/6)$ is the stability function of the two-stage Radau IIA method. We have verified numerically that the R_q stability function, based on $d = 2$, is A-acceptable for $q = 1, 2, 3, 4$. The maximum values of $|R_q(z_1, z_2)|$ are obtained for purely imaginary values of z_1 and z_2. In particular, in case $z_1 = \bar{z}_2 = t\,\mathrm{i}$ we have $z = 0$ and hence $R_q(t\,\mathrm{i}, -t\,\mathrm{i}) = R(0) = 1$ for all t. As an illustration we show in Fig. 3.1 the behaviour of $|R_q(z_1, z_2)|$ with $z_1 = z_2 = t\,\mathrm{i}$, for $q = 1, 2, 3, 4$ (which seems to be according to the numerical results the most critical situation), along with the modulus of the stability function of the Radau IIA method. □

We will now proceed with analyzing the stability for the three-dimensional situation. The starting point is again the systems defined in (3.1) and (3.2) and the matrix J is decomposed as defined in (3.7). The inner-outer iteration is now defined in (3.8) in combination with (3.9). Both for the stability analysis as well as for the actual implementation it is convenient to explicitly write out the total process. Henceforth, we will refer to this mixed, nested iteration as the (r, q)-iteration, which reads:

For $k = 1, 2, \ldots, q$: (*outer iterations*)
First stage:

$$(I_m - \gamma\tau J_1)\Delta_1^k = \tilde{D}_1^{k-1}, \tag{3.22}$$

$E_1^{k,0} = 0$,
For $j = 1, 2, \ldots, r$: (*inner iterations*)

$$\begin{aligned}&(I_m - \gamma\tau J_2)(I_m - \gamma\tau J_3)\hat{\Delta}_1^{k,j} = \Delta_1^k - (I_m - \gamma\tau(J_2 + J_3))E_1^{k,j-1},\\ &E_1^{k,j} = E_1^{k,j-1} + \hat{\Delta}_1^{k,j},\end{aligned} \tag{3.23}$$

End (for j)
$E_1^k = E_1^{k,r}$,
Second stage:

$$(I_m - \gamma\tau J_1)\Delta_2^k = \tilde{D}_2^{k-1} + L_{21}E_1^k, \tag{3.24}$$

$E_2^{k,0} = 0$,
For $j = 1, 2, \ldots, r$ (*inner iterations*)

$$\begin{aligned}&(I_m - \gamma\tau J_2)(I_m - \gamma\tau J_3)\hat{\Delta}_2^{k,j} = \Delta_2^k - (I_m - \gamma\tau(J_2 + J_3))E_2^{k,j-1},\\ &E_2^{k,j} = E_2^{k,j-1} + \hat{\Delta}_2^{k,j},\end{aligned} \tag{3.25}$$

End (for j)
$E_2^k = E_2^{k,r}$,

Stage updating:

$$\begin{aligned}&Y_{n,1}^k = Y_{n,1}^{k-1} + E_1^k + S_{12}E_2^k,\\ &Y_{n,2}^k = Y_{n,2}^{k-1} + E_2^k,\end{aligned} \tag{3.26}$$

End (for k). Set $y_{n+1}^q = Y_{n,2}^q$.
We remark that the $(1, q)$-iteration is equivalent to the mixed AMF-iteration described in (3.10) for $k = 1, 2, \ldots, q$.
The stability analysis for the (r, q)-iteration can be carried out along the same lines as given in Result 3. A rather tedious but straightforward calculation shows that the stability function $R_q(z_1, z_2, z_3)$ is given by (3.21) with M^* defined in (3.13),

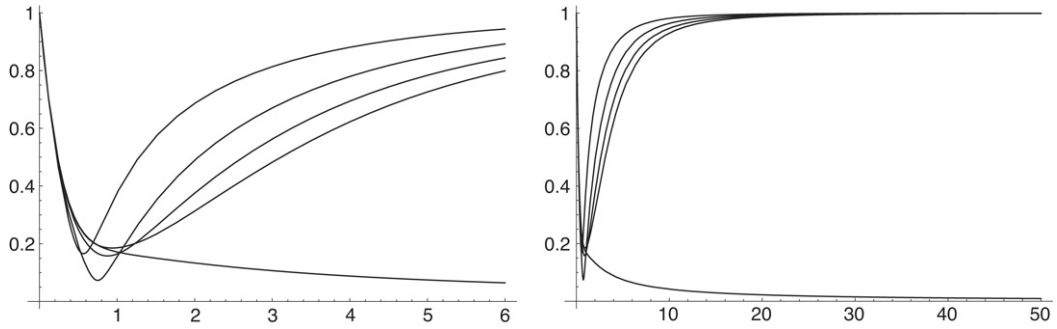

Fig. 3.2. Graphs of $|R(z_1+z_2+z_3)|$ and of $|R_q(z_1,z_2,z_3)|$ (vertical axis) for $r=1$, $q=1,2,3,4$ and $z_1=z_2=z_3=(-1+\mathrm{i})t$, $t\geq 0$ (horizontal axis). The left panel shows the situation in the neighborhood of the origin, whereas the right panel illustrates the behaviour on the interval $0\leq t\leq 50$. The situation near the origin serves to distinguish between the curves for the various q-values: the larger q, the closer $|R|$ and $|R_q|$.

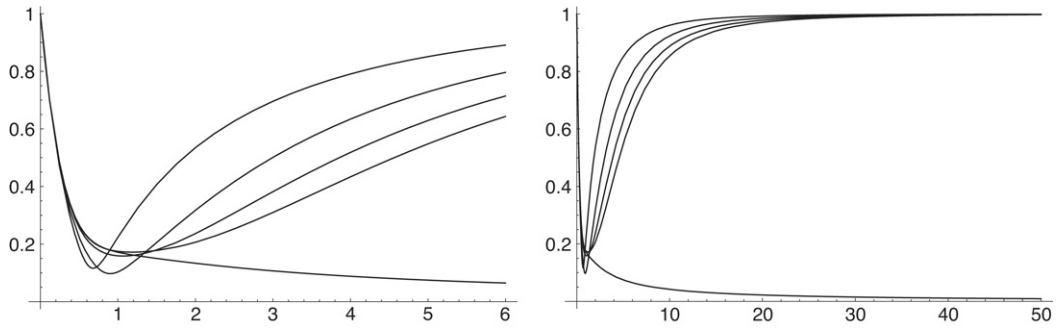

Fig. 3.3. Graphs of $|R(z_1+z_2+z_3)|$ and of $|R_q(z_1,z_2,z_3)|$ (vertical axis) for $r=2$, $q=1,2,3,4$ and $z_1=z_2=z_3=(-1+\mathrm{i})t$, $t\geq 0$ (horizontal axis). The left panel shows the situation in the neighborhood of the origin, whereas the right panel illustrates the behaviour on the interval $0\leq t\leq 50$. The situation near the origin serves to distinguish between the curves for the various q-values: the larger q, the closer $|R|$ and $|R_q|$.

but with the important difference that now x is computed from

$$\begin{aligned} x &= (1-\omega^r)^{-1}(1-\omega)(1-\gamma z_1)(1-\gamma z_2)(1-\gamma z_3),\\ \omega &= \big(\gamma z_2(1-\gamma z_2)^{-1}\big)\big(\gamma z_3(1-\gamma z_3)^{-1}\big). \end{aligned} \tag{3.27}$$

Result 4. For the three-dimensional case ($d=3$) and $q=1,2,3,4$ outer iterations, the method $y^q_{n+1}=Y^q_{n,2}$ obtained from:
(a) the $(1,q)$-iteration is A($\pi/4$)-stable;
(b) the (r,q)-iteration is A(α)-stable (with $\alpha=\pi/4$ maximal) independently of the fixed number r of inner iterations carried out.

Derivation. By using the maximum principle it follows that the maximum of $|R_q(z_1,z_2,z_3)|$ is obtained when all z_j are on the boundary of the wedge $\mathcal{W}(\alpha)$. Again, the statements (a) and (b) in Result 4 have been verified numerically. Similar to the two-dimensional case the maximum was found on the lines $z_1=z_2=z_3=-t\exp(-\mathrm{i}\alpha)$, $t\geq 0$. In the Figs. 3.2 and 3.3 we have plotted the $|R_q(z_1,z_2,z_3)|$-values for $q=1,2,3,4$ and $r=1$ and $r=2$, respectively. Here the value $\alpha=\pi/4$ has been used. As a reference, the Radau stability function $|R(z)|$ is also shown. Moreover, we have verified numerically (for all combinations of $r=1,2,\ldots,10$ and $q=1,2,3,4$) that using an α-value slightly larger than $\pi/4$ indeed yields the existence of a point $t>0$ such that $|R_q(z_1,z_2,z_3)|>1$. □

We conclude this subsection by providing some quantitative information on the values of $|R_q(z_1,z_2,z_3)|$, $q=3,4$ for the most critical situation (i.e., $z_j=(-\cos\alpha+\mathrm{i}\sin\alpha)t$, $(j=1,2,3)$ and $\alpha>\pi/4$). This information is presented in the Tables 3.1–3.3, from which we conclude that:

(i) near the origin (small t-values) the actual stability region (for each of the z_j-values) is larger than dictated by the wedge. In fact, the largest wedge contained in each of these stability regions is determined by the stiff eigenvalues ($t\to\infty$);
(ii) To gain stability it helps to increase r, the number of inner iterations;
(iii) The number of outer iterations, q, has less influence;
(iv) As we will see in Section 4, some combinations of the (r,q)-iteration with some values of the step size τ may lead to an unstable result in case of an advection dominated problem. Although increasing r will help to gain stability in such situations, it might be that a reduction of the time step is a more efficient approach.

Table 3.1
First positive t-value (with two decimal significant digits) such that $|R_q(z_1, z_2, z_3)| > 1$ for the $(1, q)$-iteration and $z_1 = z_2 = z_3 = (-\cos\alpha + \mathrm{i}\,\sin\alpha)t$.

q	$\alpha = 50°$	$\alpha = 60°$	$\alpha = 70°$	$\alpha = 80°$	$\alpha = 90°$
3	28.94	9.18	5.33	2.18	1.65
4	28.93	9.17	5.38	2.36	1.97

Table 3.2
First positive t-value (with two decimal significant digits) such that $|R_q(z_1, z_2, z_3)| > 1$ for the $(2, q)$-iteration and $z_1 = z_2 = z_3 = (-\cos\alpha + \mathrm{i}\,\sin\alpha)t$.

q	$\alpha = 50°$	$\alpha = 60°$	$\alpha = 70°$	$\alpha = 80°$	$\alpha = 90°$
3	38.47	12.18	7.12	5.08	2.94
4	38.46	12.18	7.21	5.36	2.96

Table 3.3
First positive t-value (with two decimal significant digits) such that $|R_q(z_1, z_2, z_3)| > 1$ for the (r, q)-iteration and $z_1 = z_2 = z_3 = (-\cos\alpha + \mathrm{i}\,\sin\alpha)t$ for the angles $\alpha = 50°$ and $\alpha = 90°$.

$\alpha = 50°$	q	$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$
	3	28.94	38.47	48.30	58.26	68.30
	4	28.93	38.46	48.29	58.25	68.29
$\alpha = 90°$		.				
	3	1.65	2.94	5.60	7.01	8.42
	4	1.97	2.96	6.04	7.30	8.64

4. Numerical results for a linear model problem

We will first apply the numerical procedure as described in the preceding sections to the model problem

$$u_t + a \cdot \nabla u = D\Delta u + g, \tag{4.1}$$

defined in 2 (or 3) spatial dimensions on the unit square (or unit cube). Here, as it is customary in PDEs, ∇ and Δ denote the Gradient and Laplacian operators, respectively.

At the boundaries we impose Dirichlet boundary conditions. In all tests in this section, the analytical solution is prescribed by

$$u(t, x_1, \ldots, x_d) = \cos(t^2) \prod_{i=1}^{d} x_i(1 - x_i), \quad \text{with } d = 2 \text{ or } d = 3. \tag{4.2}$$

The velocity vector $a = (a_i)$ is constant, with $a_i > 0$, and the same holds for the diffusion coefficient D. The advection and diffusion terms are discretized using symmetric, second-order stencils on a uniform grid with N internal points in each spatial direction, i.e., the mesh width is $h = 1/(N + 1)$. The inhomogeneous term g, as well as the Dirichlet boundary conditions are determined in such a way that (4.2) is the exact solution indeed. The resulting linear system has the form,

$$y' = f(t, y) := Jy + g(t), \qquad y(0) = y_0, \qquad y, f, g \in \mathbb{R}^{N^d}. \tag{4.3}$$

Here, y_0 is determined by the exact solution $u(0, x_1, \ldots, x_d)$ on the grid-points. The case $d = 2$ yields

$$J = J_1 + J_2 \quad \text{with } J_1 = I_N \otimes \tilde{J}_1, J_2 = \tilde{J}_2 \otimes I_N,$$

whereas for $d = 3$, we have

$$\begin{aligned} &J = J_1 + J_2 + J_3 \\ &\text{with } J_1 = I_N \otimes I_N \otimes \tilde{J}_1,\ J_2 = I_N \otimes \tilde{J}_2 \otimes I_N,\ J_3 = \tilde{J}_3 \otimes I_N \otimes I_N, \end{aligned} \tag{4.4}$$

with N-dimensional tridiagonal matrices $\tilde{J}_l$ $(l = 1, 2, 3)$ of the special form

$$\tilde{J}_l = \begin{pmatrix} \beta & \gamma_l & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \alpha_l & \beta & \gamma_l & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \alpha_l & \beta & \gamma_l & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & \alpha_l & \beta & \gamma_l \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \alpha_l & \beta \end{pmatrix}, \qquad \begin{cases} \alpha_l = a_l/(2h) + D/h^2 \\ \gamma_l = -a_l/(2h) + D/h^2 \\ \beta = -2D/h^2 \end{cases} \tag{4.5}$$

Table 4.1
sd-values for problem (4.1), (4.2) in 2 dimensions using iteration (3.10).

N	h	Pe	τ	$q=1$	$q=2$	$q=3$	$q=4$	→	$q=10$
32	1/33	303	3/10	1.34	1.75	1.81	1.76	→	1.75
			3/20	1.52	2.40	2.67	2.63	→	2.61
			3/40	1.72	3.14	3.61	3.51	→	3.50
			3/80	1.97	3.71	4.54	4.41	→	4.41
128	1/129	77.5	3/10	1.53	1.93	1.85	1.76	→	1.76
			3/20	1.60	2.51	2.73	2.64	→	2.62
			3/40	1.75	3.24	3.67	3.53	→	3.51
			3/80	2.00	3.83	4.58	4.43	→	4.42
512	1/513	19.5	3/10	1.66	2.10	1.91	1.82	→	1.82
			3/20	1.68	2.64	2.78	2.70	→	2.68
			3/40	1.82	3.28	3.74	3.59	→	3.57
			3/80	2.06	3.88	4.66	4.48	→	4.48

From the above splitting it is clear that all matrices J_l for the cases $d=2$ and $d=3$ have the same set of eigenvectors, respectively. For instance for the case $d=3$, the eigenvector set is given by

$$\{u_{ijk} = v_{1i} \otimes v_{2j} \otimes v_{3k},\ 1 \le i,j,k \le N\},$$

where v_{lj} denotes the jth eigenvector of the matrix $\tilde{J}_l$.

For this problem, the spatial discretization errors vanish (i.e., the PDE solution at the grid points equals the ODE solution). Hence, we only concentrate on time integration errors, which is precisely the aim of this section: to study the accuracy and convergence behaviour of the proposed time integration method. In the results presented below, the accuracy – at the end point of the integration interval – will be measured by the quantity *sd*, defined as

$$sd := -\log_{10} \|\text{numerical solution} - \text{exact solution}\|_\infty.$$

For the time interval we choose $0 \le t \le 3$. The (r,q)-iteration that we used in the tests has been described in Section 3.3 for the case of dimension $d=3$, see formulas (3.22) until (3.26). We remark that for $d=3$, the $(1,q)$-iteration coincides with the AMF-iteration described in (3.10). This also holds in the case $d=2$, by setting $J_3=0$ in (3.22)–(3.26).

Computational costs. It should be observed that the majority of the computational work in the (r,q)-iteration consists of matrix-vector products and solving linear systems with a banded matrix, the band width typically in the range 3–5. This property is independent of the number of spatial dimensions of the underlying PDE. Hence, the linear algebra work involved is much less than that encountered in fully implicit methods where 'multidimensional' systems have to be solved. Moreover, the full right-hand side function f occurring in (1.1) has to be evaluated only q times, i.e., at the start of a new single-Newton iteration. Such an f-evaluation may be quite expensive, e.g., in case of complicated diffusion terms (see the example in Section 5) or when a laborious inhomogeneous term is involved (as in the example in Section 4). Since q is usually small, this property is an advantage compared with fully explicit methods, such as stabilized Runge–Kutta methods, where in each stage the full right-hand side function f has to be re-evaluated. In Section 5 we will describe a comparison with a BDF-based code and an Runge–Kutta–Chebyshev code, including the required CPU times of all solvers.

4.1. Advection dominated case

The success of the algorithm largely depends on the position of the eigenvalues of the Jacobian of the discrete system. These eigenvalues are determined by the resolution of the spatial grid and by the ratio of advection and diffusion. A proper way to characterize a particular situation is to use the so-called cell Péclet number Pe, which is defined by $Pe = |a|h/D$ (see e.g. [14]). We will present results where we set the velocities $a_i = 1$ and the diffusion coefficient $D = 10^{-4}$, i.e., the case where advection strongly dominates diffusion. This results in $Pe = 10^4 h$ (in each spatial direction), which becomes quite large for the spatial grids that we will use. Large Péclet numbers indicate that we are dealing with the most critical situation, where the eigenvalues are close to the imaginary axis. We present results for $d=2$ and $d=3$, obtained on spatial grids with increasing resolution to see the influence on the overall performance and the convergence behaviour in particular.

For both cases we performed experiments with a constant step size τ and with a fixed number q of single-Newton iterations per step. The tables show *sd*-values for various combinations of τ and q. First, in Table 4.1, we give results for the two-dimensional problem.

From the 2D-results presented in Table 4.1 we may conclude:

- *Convergence of the iteration process.* For all values of the step size τ we observe a fast convergence. The numerical solution obtained with $q=4$ iterations is (almost) the same as the solution of the underlying Radau IIA method (column with $q=10$). This property is seen to be independent of the resolution of the spatial grid. Stopping the iteration after $q=3$ iterations yields a solution that is not yet fully converged towards the Radau IIA solution, but it is certainly of sufficient accuracy to adopt the $q=3$ result as the new step point approximation. As a matter of fact, for this problem it even shows a slightly higher precision than the Radau solution. Here we recall that the principal local error term for $q=3$ is

Table 4.2
sd-values for problem (4.1), (4.2) in 3 dimensions using the $(1, q)$-iteration.

N	h	Pe	τ	$q=1$	$q=2$	$q=3$	$q=4$	→	$q=10$
8	1/9	1111	3/10	1.75	1.98	2.07	2.12	→	1.62
			3/20	1.87	2.68	2.98	2.99	→	3.00
			3/40	2.12	3.54	3.96	3.90	→	3.90
			3/80	2.39	4.25	4.91	4.81	→	4.81
32	1/33	303	3/10	1.67	1.99	1.97	2.00	→	−3.49
			3/20	1.86	2.59	2.87	2.91	→	−12.15
			3/40	2.06	3.44	3.89	3.01	→	−5.86
			3/80	2.32	4.24	4.83	4.73	→	4.73
128	1/129	77.5	3/10	1.90	1.94	1.92	1.96	→	1.71
			3/20	1.87	2.57	2.85	2.91	→	−2.52
			3/40	2.07	3.44	3.57	1.71	→	*
			3/80	2.33	2.37	−2.90	−7.93	→	*
			3/160	2.61	−5.11	−16.13	*	→	*
			3/320	2.90	5.65	6.68	6.56	→	6.56

Table 4.3
sd-values for problem (4.1), (4.2) in 3 dimensions using the $(2, q)$-iteration.

N	h	Pe	τ	$q=1$	$q=2$	$q=3$	$q=4$	→	$q=10$
8	1/9	1111	3/10	1.70	2.00	2.12	2.14	→	2.15
			3/20	1.87	2.70	3.00	2.99	→	3.00
			3/40	2.12	3.55	3.97	3.90	→	3.90
			3/80	2.39	4.22	4.92	4.81	→	4.81
32	1/33	303	3/10	1.64	1.99	2.08	2.07	→	−3.33
			3/20	1.83	2.66	2.94	2.93	→	−9.56
			3/40	2.05	3.46	3.90	3.83	→	3.82
			3/80	2.31	4.15	4.84	4.73	→	4.73
128	1/129	77.5	3/10	1.76	2.01	2.05	2.07	→	−0.60
			3/20	1.84	2.66	2.92	2.09	→	*
			3/40	*	*	*	*	→	*
			3/80	*	*	*	*	→	*

not identical to that of the corrector. Overall, the behaviour is satisfactory and in accordance with the theoretical results concerning A-stability (see Result 3) and almost A-convergence (see Result 1).

- *Order behaviour*. The mixed iteration process has been started with the prediction (3.18). Using (2.7) we see that the order p^* after q iterations is equal to $p^* = \min(q, 3)$. This order behaviour in time is nicely observed from Table 4.1 (notice that halving the step size should yield an increase in the *sd*-value equal to $0.3p^*$).

Additionally, we repeated the above experiments (not displayed in tables) for the simpler case where $D = a_i = 1$. Hence the diffusion and advection coefficients have equal weight, resulting in much smaller Péclet numbers. We found a similar behaviour (again, the $q = 4$ solution is almost the same as the Radau solution). The only differences with Table 4.1 were: (i) $q = 4$ yielded more accurate results than $q = 3$, and (ii) for each $\{\tau; q\}$-pair, the *sd*-values on the various grids were identical. In conclusion, the Radau-based mixed iteration process (3.10) is very efficient for 2D problems, independent of the position of the eigenvalues of the discrete system.

Next, we continue with the three-dimensional version of the model problem (4.1), (4.2), with $a_i = 1$ and $D = 10^{-4}$. Here, we will employ the iteration process (3.22)–(3.26) and some results for various (r, q)-combinations are given. The Tables 4.2–4.4 show *sd*-values for $r = 1$, 2, and 5, respectively. Especially for $h = 1/33$ and $h = 1/129$ a bad convergence behaviour is observed in some cases. Initially, for small q-values (say $q \leq 3$) often useful results are obtained; however, continuing the iteration results in divergence/instability. In the tables, an asterisk denotes an *sd*-value < -20. For $h = 1/33$, it helps to continue the inner iteration process ($r = 5$); unfortunately, on the finest mesh ($h = 1/129$) several (r, q)-combinations resulted in poor performance. However, a reduction of the time step ($\tau = 3/320$ in Table 4.2) yielded satisfactory results again for those combinations. We have also added results for a coarse mesh with $h = 1/9$. Here the convergence is usually satisfactory in all cases, due to the fact that the convergence and stability regions in the neighborhood of the origin are substantially larger than indicated by the wedge (see the Tables 3.1–3.3 and the accompanying discussion). Nevertheless, the conclusion must be that the (r, q)-iteration for 3D problems with a dominating advection term must be used with some caution. For this situation the construction of a robust code needs the implementation of a variable step size strategy. This topic is subject of actual research and it is outside of the scope of the present paper. Again, these experiments seem to confirm the theory in Results 2 and 4, respectively.

For problems with substantial diffusion (compared with advection) the situation is much more favorable, in the sense that less reduction on the time step sizes is required in order to get stable and accurate solutions. For instance by applying the $(1, 3)$-iteration (or equivalently, the AMF-iteration (3.10)) to the model problem with $D = a_i = 1$, good convergence results were found as is shown in Table 4.5.

Table 4.4
sd-values for problem (4.1), (4.2) in 3 dimensions using the $(5, q)$-iteration.

N	h	Pe	τ	$q=1$	$q=2$	$q=3$	$q=4$	→	$q=10$
8	1/9	1111	3/10	1.71	2.01	2.11	2.14	→	2.15
			3/20	1.87	2.71	3.00	2.99	→	3.00
			3/40	2.12	3.56	3.97	3.90	→	3.90
			3/80	2.39	4.22	4.92	4.81	→	4.81
32	1/33	303	3/10	1.64	1.98	2.08	2.07	→	2.08
			3/20	1.83	2.67	2.94	2.93	→	2.93
			3/40	2.05	3.46	3.90	3.83	→	3.82
			3/80	2.31	4.15	4.84	4.73	→	4.73
128	1/129	77.5	3/10	1.74	2.07	2.09	2.07	→	−2.70
			3/20	1.85	2.68	2.94	1.68	→	*
			3/40	−15.93	−11.61	−9.36	−6.93	→	−2.79
			3/80	2.32	4.18	4.85	4.74	→	*

Table 4.5
sd-values for problem (4.1), (4.2) in 3 dimensions using the $(1, q)$-iteration. Here, we used $D = a_i = 1$.

N	h	Pe	τ	$q=1$	$q=2$	$q=3$	$q=4$	→	$q=10$
128	1/129	0.008	3/10	1.94	2.07	2.23	2.41	→	3.10
			3/20	2.20	2.67	3.06	3.39	→	4.01
			3/40	2.78	3.47	4.03	4.47	→	4.87
			3/80	3.29	4.20	4.98	5.53	→	5.73

Table 4.6
sd-values for problem (4.1), (4.6) in 2 dimensions using the $(1, q)$-iteration. $D = 10^{-4}$, $a_1 = a_2 = 0.2$.

N	h	Pe	τ	$q=3$	$q=5$	$q=10$
128	1/129	15.5	1/10	1.31	1.28	1.28
			1/20	2.11	2.04	2.04
			1/40	3.01	2.91	2.91
			1/80	3.92	3.80	3.80
512	1/513	3.9	1/10	1.26	1.23	1.23
			1/20	2.05	1.98	1.98
			1/40	2.95	2.84	2.85
			1/80	3.86	3.74	3.74

Non-smooth solutions. So far, we have shown the (convergence) behaviour of the AMF-approach on the basis of a smooth solution in space (cf. (4.2)). The question arises what will happen when a non-smooth solution is involved. Hundsdorfer & Verwer write in [14, p. 406]: '...the convergence of modified Newton AMF-iteration can be rather slow, especially for solutions rich in high frequencies'. To investigate that situation, we have performed an additional test for the model problem (4.1), without the inhomogeneous term $g(t)$, again on the unit square in space and t running from 0 to 1. Furthermore, $D = 10^{-4}$ and the velocities a_i are set to 0.2. Hence, this is almost a pure advection problem transporting the initial profile with constant velocity. The reason for reducing the velocities from 1 to 0.2 is that we want to keep the (steep) solution profile inside the unit square at $t = 1$. Again, we use second-order symmetric differences, both for the diffusion and the advection term. The main difference with the previous situation, however, is that we now start with the non-smooth initial field (see also [14, pp. 52–62] where a similar test for an advection model is described)

$$u(t = 0, x, y) = [\sin(\pi x)]^{100}\,[\sin(\pi y)]^{50}. \tag{4.6}$$

For this problem we do not have an analytical solution, so we first calculated a reference solution of the ODE on 2 spatial grids, i.e. $h = 1/129$ and $h = 1/513$. We restrict our considerations to 2D since the qualitative convergence behaviour caused by a potential slow damping of high-frequency modes is similar in 2D and 3D (see also Remark 2 in Section 3.2). We have tested the $(1, q)$-iteration for several q-values, to see its influence on the convergence and found the results as given in Table 4.6.

From this table we draw the following conclusions: (i) the third-order behaviour in time is nicely shown; (ii) the resolution of the spatial grid has hardly influence; (iii) the convergence is quite satisfactory: already for $q = 3$ the iteration seems to be converged.

4.2. Solving the corrector iteratively

We conclude this section by giving some results for the approach described in Section 3.1, to treat a 3D advection dominated problem. This approach consists in three nested iterations: the outer iteration is the single-Newton iteration

Table 4.7
sd-values for problem (4.1), (4.2) in 3 dimensions using the nested iteration, with q outer iterations, l middle iterations and fixed $r = 10$ inner iterations. Here, we used $D = 10^{-4}$, $a_i = 1$.

				$q=1$			$q=2$			$q=3$		
N	h	Pe	τ	$l=1$	$l=2$	$l=3$	$l=1$	$l=2$	$l=3$	$l=1$	$l=2$	$l=3$
64	1/65	153.8	3/10	1.67	1.51	1.51	2.02	1.91	1.91	2.08	2.05	2.05
			3/20	1.83	1.76	1.76	2.67	2.59	2.61	2.93	2.93	2.93
			3/40	2.05	2.02	2.02	3.48	3.42	3.42	3.91	3.91	3.91
			3/80	2.31	2.30	2.30	4.19	4.18	4.18	4.85	4.87	4.87

(2.9) to solve for the stage values in Y_n; the middle iteration process is of AMF-type and is used to solve the linear systems in (3.1); finally, the inner iteration (3.9) is used to solve the linear systems where the matrix J^* is involved.

The main idea of this nested iteration algorithm is to continue each iteration until 'convergence' to really find the Radau solution. Therefore, this approach is best implemented using an adaptive strategy where residuals have to satisfy prescribed tolerances. Such an implementation is beyond the scope of the present paper and subject of future research. To give an impression of the performance and robustness of this approach in a 3D setting, we have solved the advection dominated case with $D = 10^{-4}$ and $a_i = 1$. The nested iteration has been tested for many combinations of q, l and r, denoting, respectively, the number of outer, middle and inner iterations. It turns out that $q = 3$ outer iterations are sufficient to obtain convergence for realistic step sizes. Concerning the middle iteration process we can make the same observation. To be on the safe side, the innermost iteration process has been applied using the fixed number of $r = 10$ iterations, although in many cases smaller r-values could have been used to obtain the same accuracies. The results have been summarized in Table 4.7. This table shows that we end up with accuracies close to those of the Radau corrector itself. By comparing the results with those in the Table 4.2 until 4.4 it is evident that this approach is much more robust in the sense that it can be used for 3D problems in combination with large Pe-numbers. Therefore, we expect that this algorithm can be upgraded to an efficient and robust solver by including appropriate control mechanisms.

5. Numerical results for a nonlinear problem

Next, we continue our tests by applying the (r, q)-iteration (3.22)–(3.26) to a strongly nonlinear example, i.e. a radiation-diffusion problem from [17]. The following description and the used spatial discretization were borrowed from Ch.V of [14]. Also in [18] this problem has been used as a test example and results for an IMEX RKC scheme as well as for VODPK [5] are given in that paper. Here we will present a comparison between the results obtained with the (r, q)-iteration and the results given in [18].

The problem consists of two strongly nonlinear diffusion equations with a highly stiff reaction term (an idealization of non-equilibrium radiation diffusion in a material). The dependent variables E and T represent radiation energy and material temperature, respectively. Problems like this are for instance found in laser fusion applications. The equations are defined on the unit square for $t > 0$,

$$\begin{aligned} E_t &= \nabla \cdot (D_1 \nabla E) + \sigma (T^4 - E) \\ T_t &= \nabla \cdot (D_2 \nabla T) - \sigma (T^4 - E) \end{aligned} \tag{5.1}$$

where $\sigma = Z^3/T^3$, $D_1 = 1/(3\sigma + |\nabla E|/E)$ and $D_2 = kT^{5/2}$ with $k = 0.005$. Here, $|\nabla E|$ denotes the Euclidean norm of ∇E and $Z = Z(x, y)$ represents the atomic mass number which may vary in the spatial domain to reflect inhomogeneities in the material. We have $Z(x, y) = Z_0$ if $|x - 1/2| \le 1/6$ and $|y - 1/2| \le 1/6$ with $Z_0 \ge 1$ a constant and $Z(x, y) = 1$ otherwise. In our tests we have used $Z_0 = 10$ [17].

The initial values are constant, $E(x, y, 0) = 10^{-5}$ and $T(x, y, 0) = E(x, y, 0)^{1/4} \approx 5.62\ 10^{-2}$. As boundary conditions we have homogeneous Neumann conditions for T at all boundaries and for E at $y = 0, 1$. Further, at the left and right boundary mixed boundary conditions for E are prescribed by $\frac{1}{4}E - \frac{1}{6\sigma}E_x = 1$ at $x = 0$ and $\frac{1}{4}E + \frac{1}{6\sigma}E_x = 0$ at $x = 1$.

The solution consists of a steep (temperature) front moving to the right. For $Z_0 > 1$ the movement is hampered at the interior region with larger atomic mass number (and corresponding smaller diffusion). E is for the most part almost equal to T^4, except near the front where it is slightly larger with a steeper profile. Fig. 5.1 shows a 3D plot and contour levels of a time-accurate reference solution of T at $t = 3$ for $Z_0 = 10$, computed on a 200×200 spatial grid.

The spatial discretization is on a uniform cell centered grid with grid size h by means of second-order central conservative differencing. This gives a semi-discrete system $y'(t) = f_{diff}(y(t)) + f_{reaction}(y(t))$ of dimension $2/h^2$. At each grid point we have the nonlinear reaction system

$$f_{reaction}(E, T) = \begin{pmatrix} Z^3 T^{-3}(T^4 - E) \\ -Z^3 T^{-3}(T^4 - E) \end{pmatrix}, \qquad J_{reaction}(E, T) = \begin{pmatrix} -\alpha & \beta \\ \alpha & -\beta \end{pmatrix} \tag{5.2}$$

with $\alpha = Z^3/T^3$, $\beta = Z^3(1 + 3E/T^4)$ and eigenvalues 0 and $-(\alpha + \beta)$.

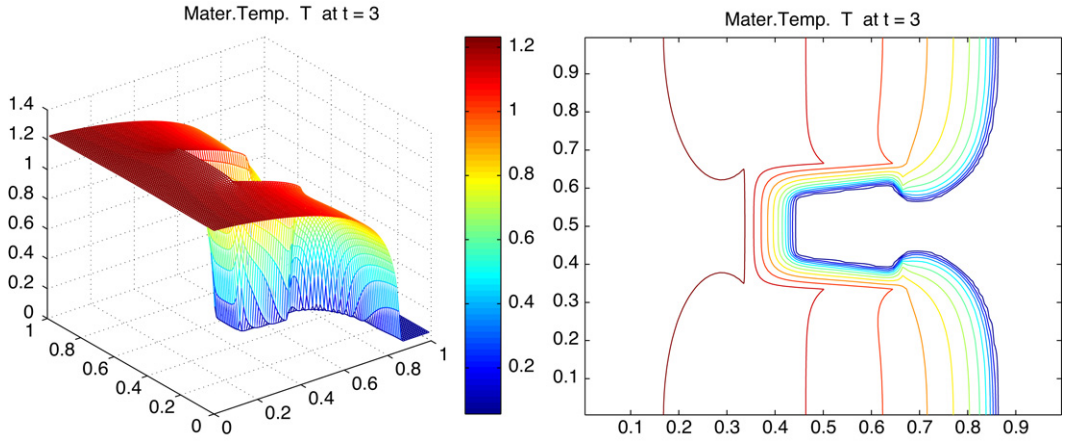

Fig. 5.1. 3D plot and contour levels of the material temperature T at time $t = 3$ for $Z_0 = 10$. Contour levels: 0.1, 0.2, . . . , 1.2.

5.1. Numerical results

5.1.1. The three solvers

We present numerical results for three different solvers:

Iterated Radau method: Here we use our (1, 3)-iteration as defined in (3.22)–(3.26). Although (5.1) is a 2D problem, we have solved this problem by using a splitting of the Jacobian into three parts: J_1 is associated with the reaction system (5.2), hence $J_1 = J_{reaction}$, whereas J_2 and J_3 represent the Jacobian matrices in x- and y-direction, respectively, of the diffusion part in the problem. As a consequence, the systems (3.22) and (3.24) involving only J_1 can be solved grid point wise. Hence, on an $N \times N$-grid, we have N^2 uncoupled systems, each of dimension 2.

The Jacobian matrices J_2 and J_3, needed in (3.23) and (3.25), have the block-triangular form

$$J_2 = \begin{pmatrix} \dfrac{\partial F^x}{\partial E} & \dfrac{\partial F^x}{\partial T} \\ \bigcirc & \dfrac{\partial G^x}{\partial T} \end{pmatrix} \quad \text{and} \quad J_3 = \begin{pmatrix} \dfrac{\partial F^y}{\partial E} & \dfrac{\partial F^y}{\partial T} \\ \bigcirc & \dfrac{\partial G^y}{\partial T} \end{pmatrix}. \tag{5.3}$$

Here, we used F and G to denote the discretized diffusion term in the first and second PDE in (5.1), respectively and the superscripts x and y refer to the spatial directions. As a simplification, we approximate D_1 by $1/(3\sigma)$, hence neglecting the ∇E-contribution. This reduces the bandwidth and leads to a block-structure in which only tridiagonal systems have to be solved. Using this approach, the systems in (3.23) and (3.25) will first solve for the T-component and subsequently for the E-component. We have also tested the method by implementing a further simplification: neglecting $\partial F^x/\partial T$ and $\partial F^y/\partial T$ in (5.3) leads to a reduction of the linear algebra work involved. However, in terms of efficiency, we found that the first strategy is to be preferred. Therefore, we will only present results based on matrices J_2 and J_3 of the form as specified in (5.3).

IMEX RKC: This solver is based on an implicit-explicit (IMEX) Runge–Kutta–Chebyshev (RKC) method, where the diffusion part is integrated by the explicit, stabilized RKC method and the stiff reaction terms (cf. (5.2)) are treated implicitly. This solver is fully described in [18] and the corresponding software is discussed in [19].[2]

VODPK: The stiff solver VODE [4,5] provided with the Krylov solver GMRES [3] with user-supplied preconditioner for solving the linear systems arising in the modified Newton iteration.[3] For this radiation-diffusion problem preconditioning is essential. Without preconditioning VODPK either fails or is very inefficient, depending on the tolerance and the grid size. We have implemented a 2×2 block-diagonal left preconditioner P which approximates the 2×2 block-diagonal of the Newton matrix. P is derived from the grid point formula

$$E'_{ij} = -\frac{4}{h^2} D_{1,ij} E_{ij} + \sigma_{ij} (T_{ij}^4 - E_{ij}), \qquad T'_{ij} = -\frac{4}{h^2} D_{2,ij} T_{ij} - \sigma_{ij} (T_{ij}^4 - E_{ij}),$$

where, similar as in our Radau-based method, D_1 is approximated by $1/(3\sigma)$. So the P_{ij}-block for grid point (x_i, y_j) reads

$$P_{ij} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + b_0 \tau \frac{4}{h^2} \begin{pmatrix} \dfrac{1}{3\sigma_{ij}} & \dfrac{T_{ij}^2 E_{ij}}{Z_{ij}^3} \\ 0 & \dfrac{7}{2} k T_{ij}^{5/2} \end{pmatrix} - b_0 \tau \begin{pmatrix} -\sigma_{ij} & Z_{ij}^3 \left(1 + 3\dfrac{E_{ij}}{T_{ij}^4}\right) \\ +\sigma_{ij} & -Z_{ij}^3 \left(1 + 3\dfrac{E_{ij}}{T_{ij}^4}\right) \end{pmatrix},$$

where b_0 is a VODPK coefficient. Note that there is no grid connectivity used in this preconditioner.

[2] http://www.netlib.org/ode/irkc.f90.

[3] http://www.netlib.org/ode/vodpk.f.

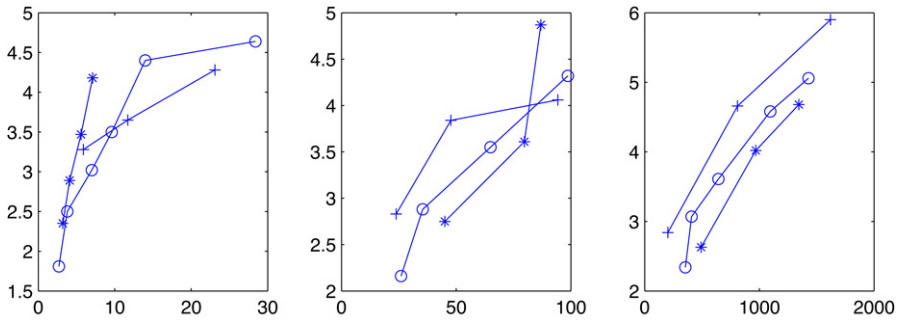

Fig. 5.2. Temporal accuracies (vertical axis), measured as $-\log_{10}(L_2\text{-errors})$ versus CPU time (horizontal axis), measured in seconds, for $h = 1/50$ (left), $h = 1/100$ (middle) and $h = 1/200$ (right). The lines marked with '+' refer to the Radau-based (1, 3)-iteration, lines with 'o' to IMEX RKC, and lines with '*' to VODPK.

5.1.2. Results

For the numerical simulations we have chosen three grid sizes, viz. $h = 1/50, 1/100, 1/200$. On each of these grids, a time-accurate reference solution has been calculated to be able to measure temporal errors. In Fig. 5.2 we show the results of the three solvers on the various grids. Here, we have plotted temporal accuracy (measured in the L_2-norm) versus CPU time, as to illustrate the efficiency of the methods. We remark that IMEX RKC and VODPK are variable step size codes, i.e., the integration process is controlled by a specified tolerance parameter. On the other hand, our Radau-based method is still in its research phase and integrates with constant step sizes. From these results we conclude that the iterated Radau method outperforms the other two solvers on the finest grid. On the grid with $h = 1/100$ the situation is similar, although less pronounced. On the coarsest mesh all three solvers show approximately equal efficiency, with slight preference for VODPK.

We anticipate that the efficiency of the iterated Radau method can be improved by adding an adequate error control strategy. Finally, we remark that VODPK did not behave very robust for this problem. We encountered many convergence failures during the integration process. Furthermore, the code only worked for rather stringent values of the tolerance parameter (see also [18] for more detailed information about the performance of VODPK).

6. Concluding remarks

We have analyzed and tested a method that is suitable to solve multi-dimensional advection-diffusion-reaction PDEs. Based on an implicit RK method of Radau IIA-type we have concentrated on a special iteration technique to solve the implicit relations that we encounter in each integration step. We have derived convergence and stability results. In a two-dimensional situation we found A(α)-convergence, with $\alpha \approx 87.9°$ and A-stability. For three-dimensional problems the situation is less favourable; we obtained A(α)-convergence with $\alpha \approx 44.7°$ and A(45°)-stability. Numerical tests with a linear problem revealed that the algorithm is still useful for advection dominated 3D problems by applying it with some care. Finally, we applied the method to a strongly non-linear, real-life problem in 2D and compared its efficiency (in terms of CPU time versus accuracies) with two existing codes, i.e., with VODPK and IMEXRKC. It turns out that the performance of the new method is at least competitive with that of the existing solvers.

Acknowledgements

The first two authors acknowledge support from the Spanish (MEC) project MTM2007-67530-C02-02; the third author acknowledges support from the Dutch national Bsik project BRICKS.

References

[1] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems, Second ed., in: Springer Series in Computational Mathematics, vol. 14, Springer, Berlin, 1996.
[2] P.N. Brown, G.D. Byrne, A.C. Hindmarsh, VODE: A variable coefficient ODE solver, SIAM J. Sci. Statist. Comput. 10 (1989) 1038–1051.
[3] Y. Saad, M. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.
[4] P.N. Brown, A.C. Hindmarsh, Reduced storage matrix methods in stiff ODE systems, J. Appl. Math. Comput. 31 (1989) 40–91.
[5] G.D. Byrne, Pragmatic experiments with Krylov methods in the stiff ODE setting, in: J. Cash, I. Gladwell (Eds.), Computational Ordinary Differential Equations, Oxford Univ. Press, Oxford, 1992, pp. 323–356.
[6] G. Dahlquist, A special stability problem for linear multistep methods, BIT 3 (1963) 27–43.
[7] J.C. Butcher, On the implementation of implicit Runge–Kutta methods, BIT 16 (1976) 237–240.
[8] J.C. Butcher, Some implementation schemes for implicit Runge–Kutta methods, in: Proceedings of the Dundee Conference on Numerical Analysis, in: Lecture Notes in Mathematics, vol. 773, 1979, pp. 12–24.
[9] G.J. Cooper, J.C. Butcher, An iteration scheme for implicit Runge–Kutta methods, IMA J. Numer. Anal. 3 (1983) 127–140.
[10] S. González–Pinto, J.I. Montijano, L. Rández, Iterative schemes for three-stage implicit Runge–Kutta methods, Appl. Numer. Math. 17 (1995) 363–382.
[11] P.J. van der Houwen, J.J.B. de Swart, Triangularly implicit iteration methods for ODE-IVP solvers, SIAM J. Sci. Statist. Comput. 18 (1997) 41–55.
[12] K. Burrage, W. Hundsdorfer, J.G. Verwer, A study of B-convergence of Runge–Kutta methods, Computing 36 (1986) 17–34.

[13] M. Calvo, S. González–Pinto, J.I. Montijano, Runge–Kutta methods for the numerical solution of stiff semilinear systems, BIT 40 (4) (2000) 611–639.
[14] W. Hundsdorfer, J.G. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, in: Springer Series in Computational Mathematics, vol. 33, Springer, Berlin, 2003.
[15] P.J. van der Houwen, B.P. Sommeijer, Approximate factorization for time-dependent partial differential equations, J. Comput. Appl. Math. 128 (2001) 447–466.
[16] W. Hundsdorfer, A note on stability of the Douglas splitting method, Math. Comp. 67 (1998) 183–190.
[17] V.A. Mousseau, D.A. Knoll, W.J. Rider, Physics-based preconditioning and the Newton–Krylov method for non-equilibrium radiation diffusion, J. Comput. Phys. 160 (2000) 743–765.
[18] J.G. Verwer, B.P. Sommeijer, An implicit-explicit Runge–Kutta–Chebyshev scheme for diffusion–reaction equations, SIAM J. Sci. Comput. 25 (2004) 1824–1835.
[19] L.F. Shampine, B.P. Sommeijer, J.G. Verwer, IRKC: An IMEX solver for stiff diffusion-reaction PDEs, J. Comput. Appl. Math. 196 (2006) 485–497.