Journal of Computational and Applied Mathematics 286 (2015) 196–210

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Application of approximate matrix factorization to high order linearly implicit Runge–Kutta methods

Hong Zhang *, Adrian Sandu, Paul Tranquilli

Computational Science Laboratory, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, United States

ARTICLE INFO

Article history:
Received 17 August 2014
Received in revised form 2 March 2015

MSC:
65L04
65L06
65M20

Keywords:
Approximate matrix factorization
Linearly implicit Runge–Kutta methods
High order
Reaction–diffusion equations

ABSTRACT

Linearly implicit Runge–Kutta methods with approximate matrix factorization can solve efficiently large systems of differential equations that have a stiff linear part, e.g. reaction–diffusion systems. However, the use of approximate factorization usually leads to loss of accuracy, which makes it attractive only for low order time integration schemes. This paper discusses the application of approximate matrix factorization with high order methods; an inexpensive correction procedure applied to each stage allows to retain the high order of the underlying linearly implicit Runge–Kutta scheme. The accuracy and stability of the methods are studied. Numerical experiments on reaction–diffusion type problems of different sizes and with different degrees of stiffness illustrate the efficiency of the proposed approach.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

A frequently used approach to solve partial differential equations (PDEs) is the method of lines, where the spatial derivative terms are discretized first using techniques such as finite differences, finite volumes or finite elements, and then integrating the resulting system of ordinary differential equations (ODEs) in time. Discretization of PDEs with linear terms in space leads to a semi-linear ODE system of the form

$$y' = F(y,t) = \mathbf{L}y + f(y,t), \quad y \in \mathbb{R}^N \tag{1}$$

where $\mathbf{L}$ is a spatial linear operator and $f(y,t)$ is nonlinear. We consider the case where the linear term has a fast characteristic time scale and the nonlinear term has a slow characteristic time scale. Due to the Courant–Friedrichs–Levy (CFL) condition, the step size of an explicit time integrator is restricted by the fastest time scale. To alleviate the time step constraint imposed by the stiff linear term with a reasonably low computational cost, linearly implicit methods (particular cases of implicit–explicit methods) treat the stiff linear term implicitly while the nonlinear term explicitly.

Various families of linearly implicit integration methods have been proposed and successfully applied to solve PDEs with linear dispersion and dissipation [1–4]. Fully implicit schemes solve at each step a linear system where the matrix involves the Jacobian of the right hand side function. Efficient schemes specialized in the solution of (1) such as linearly implicit methods use only the part of the Jacobian associated with the stiff linear term, resulting in linear systems of the form

$$(\mathbf{I} - h\gamma\,\mathbf{L})\,x = \ell, \quad x, \ell \in \mathbb{R}^N, \tag{2}$$

* Corresponding author.
E-mail addresses: zhang@vt.edu (H. Zhang), sandu@cs.vt.edu (A. Sandu), ptranq@vt.edu (P. Tranquilli).

http://dx.doi.org/10.1016/j.cam.2015.03.005
0377-0427/© 2015 Elsevier B.V. All rights reserved.

where h is the step size, γ is a parameter of the integration scheme, and the right hand side ℓ is determined by the method.

The matrix $\mathbf{L}$ is usually sparse but has a large bandwidth, especially when high order spatial discretization schemes are applied. Consequently the LU factorization of the matrix in (2) can be very costly in large scale problems. One approach to increase efficiency is to apply iterative solvers to (2), as in [5], but there are associated challenges related to preconditioning and convergence.

An alternative approach to increase the computational efficiency is approximate matrix factorization (AMF). Assuming that the matrix is a sum of simpler matrices

$$\mathbf{L} = \sum_{r=1}^{R} \mathbf{L}_r \tag{3}$$

AMF replaces the system matrix (2) with a product of simpler, and easier to factorize, matrices

$$\mathbf{I} - h\,\gamma\,\mathbf{L} \approx \mathbf{I} - h\,\gamma\,\widetilde{\mathbf{L}} = \prod_{r=1}^{R} (\mathbf{I} - h\,\gamma\,\mathbf{L}_r)\,. \tag{4}$$

The approximation formula (4) defines implicitly the matrix $\widetilde{\mathbf{L}}$ as

$$\widetilde{\mathbf{L}} = \mathbf{L} + \sum_{k=2}^{R} (-h\gamma)^{k-1} \sum_{1 \le i_1 < i_2 < \cdots < i_k \le R} \mathbf{L}_{i_1}\,\mathbf{L}_{i_2} \cdots \mathbf{L}_{i_k}. \tag{5}$$

For example, consider $\mathbf{L}$ to be the discrete two-dimensional Laplace operator. It can be written in the form (3) where $\mathbf{L}_1$ and $\mathbf{L}_2$ correspond to the derivatives along the x direction and y direction, respectively. The AMF approximation corresponds to the alternating directions factorization [6,7]

$$\mathbf{I} - h\,\gamma\,\widetilde{\mathbf{L}} := (\mathbf{I} - h\,\gamma\mathbf{L}_1)\;(\mathbf{I} - h\,\gamma\mathbf{L}_2)$$

where

$$\widetilde{\mathbf{L}} = \mathbf{L} - h\,\gamma\,\mathbf{L}_1\,\mathbf{L}_2. \tag{6}$$

The idea of using AMF to speed up calculations in implicit time integration has appeared multiple times in the literature. Sandu [8] discussed a family of methods named ELADI, which are Rosenbrock-W schemes that make use of AMF to speed up calculations. The order of the resulting discretization remains unchanged since Rosenbrock-W schemes can accommodate any Jacobian approximation. Houwen et al. provided a survey for AMF methods applied in the context of several different linear integration schemes, and provide stability results for such schemes [9]. The discussion is limited to methods of low order, two and three, presumably due to the inaccurate nature of the AMF approximation. Gonzalez [10] proposed a way to apply AMF-refinements to second-order and third-order Rosenbrock-type methods for solving advection–diffusion–reaction PDEs. But their methods are limited to the low or medium precision level, and no generalization to higher order is supplied. In [11] Beck et al. compared the efficiency of AMF versus Krylov based approaches to the solution of linear systems in the context of Newton iterations arising in Radau [12] and Peer [13] integration methods. These methods avoid the issue of order degradation, through the use of integration schemes in which the Jacobian of the spatial discretization does not appear explicitly. They conclude that AMF methods are extremely efficient, particularly when low accuracy solutions are sought. Berzins et al. [14,15] presented a method for solving the linear system in a Newton iteration arising from several classes of time integration methods including theta methods, backward differential formulas, and implicit–explicit (IMEX) multistep methods. They performed an analysis of the error arising from operator splitting and provided a method to control timesteps such as to guarantee Newton convergence when using AMF. Since AMF is only used to speed up the solution of the nonlinear equations, the error does not affect the order of accuracy at the time stepping level. Calvo and Gerisch [16] applied AMF to a form of linearly implicit Runge–Kutta (LIRK) methods that avoids the computation of matrix–vector products. First order of convergence is obtained by using a third-order LIRK method, and improved to second order by adding a correction to the solution at each time step.

None of the existing methods that take advantage of AMF can provide highly accurate results and there is still room for improvement in efficiency. The goal of this work is to achieve high accuracy while maintaining a low computational cost. The focus is on using AMF with linearly implicit Runge–Kutta methods of high order. The main contribution of this paper is to account for the inherent inaccuracy of AMF through low cost refinements of the stage values and to recover the accuracy of the underlying time discretization.

The remainder of this paper is organized as follows. Section 2 introduces LIRK methods and the existing approaches to apply AMF. Section 3 presents a new strategy to incorporate AMF by using an inexpensive stage refinement procedure. An error analysis explains how this strategy solves the accuracy degradation issue that affects existing approaches. Stability issues are also investigated. Section 4 reports numerical results for a variety of test problems of different dimensions and different degrees of stiffness, and illustrates the convergence behavior and efficiency of the approach. Conclusions are drawn in Section 5.

2. Linearly implicit Runge–Kutta methods

A general linearly implicit Runge–Kutta (LIRK) scheme proposed by Calvo, de Frutos, and Novo [2] is obtained by applying the IMEX Runge–Kutta methods [17,18]

$$Y_1 = y_n, \tag{7a}$$

$$Y_i = y_n + h \sum_{j=1}^{i-1} a_{i,j} f(Y_j, t_n + c_j h) + h \sum_{j=1}^{i} \widehat{a}_{i,j}\, g(Y_j, t_n + c_j h), \quad 2 \le i \le s, \tag{7b}$$

$$y_{n+1} = y_n + h \sum_{j=1}^{s} b_j f(Y_j, t_n + c_j h) + h \sum_{j=1}^{s} \widehat{b}_j\, g(Y_j, t_n + c_j h), \tag{7c}$$

to solve (1) where the stiff component is linear, $g(y) := \mathbf{L}y$:

$$Y_1 = y_n, \tag{8a}$$

$$\left(\mathbf{I} - h\widehat{a}_{i,i}\,\mathbf{L}\right)\, Y_i = \ell_i := y_n + h \sum_{j=1}^{i-1} a_{i,j} f(Y_j, t_n + c_j h) + h \sum_{j=1}^{i-1} \widehat{a}_{i,j}\,\mathbf{L}\, Y_j, \quad 2 \le i \le s, \tag{8b}$$

$$y_{n+1} = y_n + h \sum_{j=1}^{s} b_j f(Y_j, t_n + c_j h) + h \sum_{j=1}^{s} \widehat{b}_j \mathbf{L}\, Y_j. \tag{8c}$$

Order conditions for LIRK methods are derived in [2], and simplifications of the IMEX Runge–Kutta conditions are possible due to the special form of the nonlinear term. For notational brevity, the range of index i will be $2 \le i \le s$ for the remainder of this paper if not specified otherwise.

The coefficients $\widehat{a}_{i,i}$ in practical methods are chosen to be equal to the same value γ for computational efficiency, as this allows to reuse the same LU factorization in all stages. A linear transformation of variables allows for a reformulation of the stage equations (8b) in a form that avoids explicit multiplications by the matrix $\mathbf{L}$

$$(\mathbf{I} - h\,\gamma\,\mathbf{L})\; U_i = y_n + \sum_{j=1}^{i-1} \frac{\widehat{a}_{i,j} - a_{i,j}}{\gamma}\; Y_j + h \sum_{j=1}^{i-1} a_{i,j} F(Y_j, t_n + c_j h), \tag{9a}$$

$$Y_i = U_i - \sum_{j=1}^{i-1} \frac{\widehat{a}_{i,j} - a_{i,j}}{\gamma}\; Y_j. \tag{9b}$$

Moreover, it is convenient to choose pairs of methods with the same weights $b_j = \widehat{b}_j$ in which case the next step solution (8c) is

$$y_{n+1} = y_n + h \sum_{j=1}^{s} b_j F(Y_j, t_n + c_j h). \tag{9c}$$

Calvo and Gerisch [16] studied the use of LIRK methods with AMF. The approximation is obtained by replacing $\mathbf{I} - h\gamma\mathbf{L}$ with $\mathbf{I} - h\gamma\widetilde{\mathbf{L}}$ in (9a), or equivalently, by using the matrix $\widetilde{\mathbf{L}}$ instead of $\mathbf{L}$ in (8)

$$\widetilde{Y}_1 = y_n,$$

$$\left(\mathbf{I} - h\,\gamma\,\widetilde{\mathbf{L}}\right)\, \widetilde{Y}_i = y_n + h \sum_{j=1}^{i-1} a_{i,j} f(\widetilde{Y}_j, t_n + c_j h) + h \sum_{j=1}^{i-1} \widehat{a}_{i,j}\,\widetilde{\mathbf{L}}\,\widetilde{Y}_j,$$

$$\widetilde{y}_{n+1} = y_n + h \sum_{j=1}^{s} b_j f(\widetilde{Y}_j, t_n + c_j h) + h \sum_{j=1}^{s} \widehat{b}_j\,\widetilde{\mathbf{L}}\,\widetilde{Y}_j.$$

It can also be regarded as a direct application of the LIRK method (8) to solve the perturbed ODE system

$$\widetilde{y}' = \widetilde{\mathbf{L}}\widetilde{y} + f(\widetilde{y}, t) = F(\widetilde{y}, t) - \left(\mathbf{L} - \widetilde{\mathbf{L}}\right)\, \widetilde{y} \tag{10}$$

instead of the original system (1). The first-order behavior of this approach has been explained in [16] by the fact that the perturbation added in (10) changes the solution over one time step by $\mathcal{O}(h^2)$.

To recover second order Calvo and Gerisch [16] apply corrections to the numerical solution obtained by LIRK with AMF. One such correction has the form

$$y_{n+1} = \widetilde{y}_{n+1} + h\, \left(\mathbf{I} - h\,\gamma\,\widetilde{\mathbf{L}}\right)^{-1} \left(\mathbf{L} - \widetilde{\mathbf{L}}\right)\, y_n. \tag{11}$$

The matrix inverse in the correction term uses the same LU factorization as the solution. The presence of the matrix inverse in the correction term ensures the linear stability of the new solution (11). It is also noted in [16] that "regaining order three using corrections similar to (11) is not feasible with a computational cost comparable with the cost of recovering order two".

3. LIRK–AMF methods with stage refinement

We consider a different way to incorporate AMF into LIRK methods, which forms the basis of all approaches presented in this paper. In order to keep the right-hand side of the original ODE system (1), we only approximate $\mathbf{I} - h\gamma\mathbf{L}$ with $\mathbf{I} - h\gamma\widetilde{\mathbf{L}}$ when computing the Runge–Kutta stages:

$$\widetilde{Y}_1 = \widetilde{y}_n, \tag{12a}$$

$$\left(\mathbf{I} - h\gamma\widetilde{\mathbf{L}}\right)\widetilde{Y}_i = \widetilde{\ell}_i := \widetilde{y}_n + h\sum_{j=1}^{i-1} a_{i,j} f(\widetilde{Y}_j, t_n + c_j h) + h\sum_{j=1}^{i-1} \widehat{a}_{i,j}\mathbf{L}\widetilde{Y}_j, \tag{12b}$$

$$\widetilde{y}_{n+1} = \widetilde{y}_n + h\sum_{j=1}^{s} b_j f(\widetilde{Y}_j, t_n + c_j h) + h\sum_{j=1}^{s} \widehat{b}_j\mathbf{L}\widetilde{Y}_j. \tag{12c}$$

Thus the new method uses an inexact Jacobian for the implicit part in the LIRK scheme. The change of the left hand side in (12b) will affect the solution, and a correction is needed to restore accuracy.

Consider the solution of the original stage equation (8b)

$$(\mathbf{I} - h\gamma\mathbf{L})\ Y_i - \ell_i = 0$$

by simplified Newton iterations of the form:

$$Y_i^{(k)} = Y_i^{(k-1)} - \left(\mathbf{I} - h\gamma\widetilde{\mathbf{L}}\right)^{-1}\cdot\left((\mathbf{I} - h\gamma\mathbf{L})\ Y_i^{(k-1)} - \ell_i\right). \tag{13}$$

For example, the direct solution of (12b) is:

$$Y_i^{(0)} = \left(\mathbf{I} - h\gamma\widetilde{\mathbf{L}}\right)^{-1}\ \ell_i,$$

and the solution after one refinement iteration is:

$$\begin{aligned} Y_i^{(1)} &= Y_i^{(0)} - \left(\mathbf{I} - h\gamma\widetilde{\mathbf{L}}\right)^{-1}\cdot\left((\mathbf{I} - h\gamma\mathbf{L})\ Y_i^{(0)} - \ell_i\right) \\ &= Y_i^{(0)} - \left(\mathbf{I} - h\gamma\widetilde{\mathbf{L}}\right)^{-1}\cdot\left(h\gamma\widetilde{\mathbf{L}} - h\gamma\mathbf{L}\right)\ Y_i^{(0)}. \end{aligned} \tag{14}$$

Next we analyze the linear system solution errors and investigate how this errors propagate to affect the solution at the next step.

3.1. Error analysis

Consider the exact stage solution

$$Y_i = (\mathbf{I} - h\gamma\mathbf{L})^{-1}\ \ell_i.$$

The linear system solution error after k iterations is defined as

$$\varepsilon_i^{(k)} = Y_i^{(k)} - Y_i.$$

From (13) we obtain

$$\varepsilon_i^{(k)} = \varepsilon_i^{(k-1)} - \left(\mathbf{I} - h\gamma\widetilde{\mathbf{L}}\right)^{-1}\cdot(\mathbf{I} - h\gamma\mathbf{L})\ \varepsilon_i^{(k-1)} \tag{15a}$$

$$= \left(\mathbf{I} - h\gamma\widetilde{\mathbf{L}}\right)^{-1}\cdot\left(\mathbf{I} - h\gamma\widetilde{\mathbf{L}} - (\mathbf{I} - h\gamma\mathbf{L})\right)\ \varepsilon_i^{(k-1)} \tag{15b}$$

$$= -\left(\mathbf{I} - h\gamma\widetilde{\mathbf{L}}\right)^{-1}\cdot\left(h\gamma\widetilde{\mathbf{L}} - h\gamma\mathbf{L}\right)\ \varepsilon_i^{(k-1)}. \tag{15c}$$

Nonstiff or moderately stiff case. In the nonstiff or moderately stiff case we have $\|h\mathbf{L}_i\| = \mathcal{O}(h)$, therefore $\|\widetilde{\mathbf{L}} - \mathbf{L}\| = \mathcal{O}(h)$ and

$$\left\|\left(\mathbf{I} - h\gamma\widetilde{\mathbf{L}}\right)^{-1}\cdot\left(h\gamma\widetilde{\mathbf{L}} - h\gamma\mathbf{L}\right)\right\| = \mathcal{O}\left(h^2\right).$$

Consequently from (15c) the error decrease is

$$\left\|\varepsilon_i^{(k+1)}\right\| = \mathcal{O}(h^2)\ \left\|\varepsilon_i^{(k)}\right\| \Rightarrow \left\|\varepsilon_i^{(k)}\right\| = \mathcal{O}\left(h^{2k+2}\right).$$

Highly stiff case. For the highly stiff case $\|h\mathbf{L}_i\| \gg 1$. We make the assumption that, for any $\bar{h}$ there exists $0 < \rho(\bar{h}) < 1$ such that the following matrix norm is uniformly bounded for any step size smaller than $\bar{h}$:

$$\left\|\left(\mathbf{I} - h\gamma\widetilde{\mathbf{L}}\right)^{-1}\cdot\left(h\gamma\widetilde{\mathbf{L}} - h\gamma\mathbf{L}\right)\right\| \le \rho(\bar{h}) < 1, \quad \forall h : 0 \le h \le \bar{h}.$$

In the highly stiff case the error decrease equation (15c) leads to

$$\left\| \varepsilon_i^{(k)} \right\| \le \rho \left\| \varepsilon_i^{(k-1)} \right\| \Rightarrow \left\| \varepsilon_i^{(k)} \right\| \le \rho^k \, \varepsilon_i^{(0)}.$$

We expect that the convergence rate will decrease with increasing step sizes, i.e., $\rho(\overline{h}) \to 1$ when $\overline{h} \to \infty$.

Example 3.1 (*Dimensional Splitting of the Discrete Diffusion Operator on a Cartesian Grid*)**.** Consider the two-dimensional diffusion operator with periodic boundary conditions on a domain of size $L_X \times L_Y$. It is discretized on an $M \times N$ grid of size Δx, Δy. We perform a dimensional splitting. The error equation (15c) can be written as

$$(\mathbf{I} - h\,\gamma\,\mathbf{L}_1)\;(\mathbf{I} - h\,\gamma\,\mathbf{L}_2)\;\varepsilon_i^{(k)} = (h\,\gamma\,\mathbf{L}_1)\;(h\,\gamma\,\mathbf{L}_2)\;\varepsilon_i^{(k-1)}. \tag{16}$$

Consider the discrete frequencies

$$-\frac{M}{2} \le m \le \frac{M}{2} - 1, \qquad -\frac{N}{2} \le n \le \frac{N}{2} - 1, \qquad \tilde{m} = \frac{2\pi m}{L_X}, \qquad \tilde{n} = \frac{2\pi n}{L_Y}.$$

A discrete Fourier transform applied to (16) gives the following error equation for each spatial mode (m, n) of the error:

$$(1 + h\,\gamma\,\tilde{m}^2)(1 + h\,\gamma\,\tilde{n}^2)\,\hat{\varepsilon}_{m,n}^{(k)} = (-h\,\gamma\,\tilde{m}^2)(-h\,\gamma\,\tilde{n}^2)\,\hat{\varepsilon}_{m,n}^{(k-1)}.$$

Let

$$z_1 = \frac{h}{\Delta x^2}\left(\frac{2\pi m}{M}\right)^2, \qquad z_2 = \frac{h}{\Delta y^2}\left(\frac{2\pi n}{N}\right)^2.$$

The evolution of the mode (m, n) of the error is

$$\hat{\varepsilon}_{m,n}^{(k+1)} = \frac{(\gamma z_1)(\gamma z_2)}{(1 + \gamma z_1)(1 + \gamma z_2)}\,\hat{\varepsilon}_{m,n}^{(k)}.$$

The error amplification factor for the (m, n) mode is

$$R_{m,n} = \frac{(\gamma z_1)(\gamma z_2)}{(1 + \gamma z_1)(1 + \gamma z_2)}, \qquad |R_{m,n}| < 1, \qquad |R_{m,n}| \overset{z_1, z_2 \to \infty}{\longrightarrow} 1.$$

Therefore we expect that more iterations will be required for stiff problems. The AMF with correction will work well for mildly stiff problems. It will work well for stiff problems only when the solution is smooth, and the high order modes are approaching zero. Similar conclusions are drawn for the three-way splitting of a three dimensional diffusion problem

$$R_{m,n,k} = \frac{(\gamma z_1)(\gamma z_2) + (\gamma z_1)(\gamma z_3) + (\gamma z_2)(\gamma z_3) + (\gamma z_1)(\gamma z_2)(\gamma z_3)}{(1 + \gamma z_1)(1 + \gamma z_2)(1 + \gamma z_3)}.$$

Remark 3.1. The accuracy analysis in Calvo and Gerisch's paper [16] considers the non-stiff case. For very stiff systems $\left(\mathbf{I} - h\,\gamma\,\widetilde{\mathbf{L}}\right)^{-1}$ in the correction term (11) is not a good approximation to the identity matrix, as in [16, Remark 1], since $\widetilde{\mathbf{L}}$ dominates. In this case the corrected solution has an order lower than two.

3.2. Propagation of linear system errors

The computation of stage values via (12) and (13) propagates the linear system errors from one stage to another. To account for the total error consider the methods (8) and (12) and let

$$\delta Y_i = \widetilde{Y}_i - Y_i. \tag{17}$$

We assume that these errors are small. The exact stage equations (8b) read

$$(\mathbf{I} - h\,\gamma\,\mathbf{L})\;Y_i = y_n + h\sum_{j=1}^{i-1} a_{i,j}\,f(Y_j, t_n + c_j h) + h\sum_{j=1}^{i-1} \widehat{a}_{i,j}\,\mathbf{L}\,Y_j.$$

The AMF stage equations are solved inexactly and read

$$(\mathbf{I} - h\,\gamma\,\mathbf{L})\;\left(\widetilde{Y}_i - \varepsilon_i\right) = y_n + h\sum_{j=1}^{i-1} a_{i,j}\,f(\widetilde{Y}_j, t_n + c_j h) + h\sum_{j=1}^{i-1} \widehat{a}_{i,j}\,\mathbf{L}\,\widetilde{Y}_j,$$

where ε_i is the error due to the simplified Newton approximation of the system solution. Using (17) we express this in terms of the solution of the exact stages

$$(\mathbf{I} - h\gamma\,\mathbf{L})\ (Y_i + \delta Y_i - \varepsilon_i) = y_n + h\sum_{j=1}^{i-1} a_{i,j} f(Y_j, t_n + c_j h) + h\sum_{j=1}^{i-1} \widehat{a}_{i,j}\,\mathbf{L}\,Y_j + h\sum_{j=1}^{i-1} a_{i,j} f_j'\,\delta Y_j + h\sum_{j=1}^{i-1} \widehat{a}_{i,j}\,\mathbf{L}\,\delta Y_j,$$

where we use the mean value theorem

$$f(Y_j + \delta Y_j, t_n + c_j h) - f(Y_j, t_n + c_j h) = f_j' \cdot \delta Y_j, \qquad f_j' = \int_0^1 f_y(Y_j + s\,\delta Y_j, t_n + c_j h)\,ds.$$

The nonstiff/moderately stiff assumption about the nonlinear terms f implies that these average Jacobians are of moderate size,

$$\|f_j'\| = \mathcal{O}(1), \quad \forall j. \tag{18}$$

After subtracting the exact stage equations we are left with the error relation

$$\delta Y = \left(\mathbf{I} - h\widehat{\mathbf{A}} \otimes \mathbf{L} - h\,\mathbf{A} \odot F'\right)^{-1} \left(\mathbf{I} - h\,\widehat{\boldsymbol{\Gamma}} \otimes \mathbf{L}\right)\ \varepsilon \tag{19}$$

where $\widehat{\boldsymbol{\Gamma}} = \operatorname{diag}(\widehat{\mathbf{A}})$, $(\mathbf{A} \odot F')_{i,j} = a_{i,j} f_j'$ and the symbol $\otimes$ denotes the tensor product. For nonstiff or moderately stiff problems $\|h\mathbf{L}\| = \mathcal{O}(h)$, $\|hF'\| = \mathcal{O}(h)$, and for small step sizes we have

$$\|\delta Y\| = (1 + \mathcal{O}(h))\ \|\varepsilon\|.$$

For highly stiff systems $\|h\mathbf{L}\| \to \infty$ and we have

$$\delta Y = \left(\widehat{\mathbf{A}} \otimes (\mathbf{L}/\|\mathbf{L}\|)\right)^{-1} \left(\widehat{\boldsymbol{\Gamma}} \otimes (\mathbf{L}/\|\mathbf{L}\|)\right)\ \varepsilon$$

therefore

$$\|\delta Y\| = \mathcal{O}(1)\ \|\varepsilon\|.$$

In both the nonstiff and the stiff cases the stage error is of the size of the linear system solution error.

From the exact step equation the error in the solution is

$$\delta y_{n+1} = h\sum_{j=1}^{s} b_j f_j'\,\delta Y_j + h\sum_{j=1}^{s} \widehat{b}_j \mathbf{L}\,\delta Y_j.$$

Non-stiff or moderately stiff problems. In the non-stiff or moderately stiff case where $\|\mathbf{L}\| = \mathcal{O}(1)$ we have

$$\varepsilon \sim \mathcal{O}\left(h^{2k+2}\right) \Rightarrow \delta Y \sim \mathcal{O}\left(h^{2k+2}\right) \Rightarrow \|\widetilde{y}_{n+1} - y_{n+1}\| \sim \mathcal{O}\left(h^{2k+3}\right).$$

For $k = 0$ and $k = 1$ correction iterations we have the following results.

Theorem 3.1. *If a LIRK method of order higher than 2 is applied to a nonstiff or moderately stiff case of (1) with the AMF technique according to (12), then the order of the method will reduce to second order.*

Theorem 3.2. *If a LIRK method of order 3 or 4 is applied to a nonstiff or moderately stiff case of (1) with the AMF technique according to (12), and one correction iteration (14) is applied to each stage value Y_i, the order of the method is the same as that of the original method.*

Remark 3.2. Taking more iterations in the correction procedure may further reduce the linear system solution errors.

Remark 3.3. Correspondingly two iterations of the correction procedure are needed for a LIRK method of order 5 or 6 since $k = 2$ yields an error in the solution of magnitude $\mathcal{O}\left(h^7\right)$. The idea can be extended to arbitrarily high order methods.

Very stiff problems. In the highly stiff case a more complex analysis based on (19) is called for since $\|h\mathbf{L}\|$ can be large and $\|h\mathbf{L} \cdot \delta Y_j\|$ can also become large. We consider LIRK methods with a stiffly accurate implicit component $\widehat{b}_i = \widehat{a}_{s,i}$. We have that

$$\begin{aligned} Y_s &= y_n + h\sum_{j=1}^{i-1} a_{s,j} f(Y_j, t_n + c_j h) + h\sum_{j=1}^{s} \widehat{a}_{s,j}\,\mathbf{L}\,Y_j, \\ y_{n+1} &= Y_s + h\sum_{j=1}^{s} (b_j - a_{s,j}) f(Y_j, t_n + c_j h). \end{aligned}$$

The corresponding error equation

$$\delta y_{n+1} = \delta Y_s + h\sum_{j=1}^{s} (b_j - a_{s,j}) f_j'\,\delta Y_j$$

and the non-stiff condition (18) reveal that the error in the solution is of the same size as the error in the linear solvers

$$\|\widetilde{y}_{n+1} - y_{n+1}\| \sim \|\delta Y\| \sim \|\varepsilon\|.$$

3.3. Stability considerations

Following Calvo and Gerisch [16] we perform stability analysis using the following scalar test problem:

$$hf(y) = iwy, \qquad h\mathbf{L} = z = z_1 + z_2, \qquad h\widetilde{\mathbf{L}} = z_1 + z_2 - \gamma z_1 z_2. \tag{20}$$

The LIRK method (8) applied to the test problem (20) gives:

$$\begin{aligned} &y_{n+1} = R(z, iw)\, y_n, \\ &R(z, iw) = 1 + (iw\, b + z\widehat{b})^T \left(\mathbf{I} - z\widehat{A} - iwA\right)^{-1} \mathbf{1}, \\ &R(\infty, iw) = 1 - \widehat{b}^T \widehat{A}^{-1} \mathbf{1}. \end{aligned}$$

The LIRK + AMF method (12) applied to the test problem (20) gives:

$$\begin{aligned} &\widetilde{Y}_i = \left((1+\gamma^2 z_1 z_2)\mathbf{I} - z\widehat{A} - iwA\right)^{-1} \mathbf{1}\, y_n, \\ &\widetilde{y}_{n+1} = y_n + iw \sum_{j=1}^{s} b_j \widetilde{Y}_j + (z_1 + z_2) \sum_{j=1}^{s} \widehat{b}_j \widetilde{Y}_j. \end{aligned}$$

Consequently, for very stiff linear components the overall scheme is weakly stable:

$$\begin{aligned} &y_{n+1} = R(z_1, z_2, iw)\, y_n, \\ &R(z_1, z_2, iw) = 1 + (iw\, b + z\widehat{b})^T \left((1+\gamma^2 z_1 z_2)\mathbf{I} - z\widehat{A} - iwA\right)^{-1} \mathbf{1}, \\ &R(\infty, \infty, iw) = 1. \end{aligned}$$

Fig. 1 shows the stability regions of two LIRK methods with AMF (see Section 4 for details) in the volume $(z1, z2, \omega)$. By limiting the variable ω to $[-1, 1]$ and computing numerically

$$\rho(z_1, z_2) = \max_{-1 \le \omega \le 1} |R(z_1, z_2, \omega)|,$$

we present their stability regions in the plane (z_1, z_2) with contour plots in Figs. 2 and 3. It can be observed that, as z_1 and z_2 go from 0 to $-\infty$, the values first decrease, then increase, approaching 1. This agrees to the conclusion of our theoretical analysis.

For the scheme with one refinement step we have:

$$\begin{aligned} &\tau = \frac{1 - \gamma\,(z_1 + z_2) + 2\,\gamma^2 z_1 z_2}{(1 - \gamma z_1)\,(1 - \gamma z_2)}, \\ &\widetilde{Y} = \left(\tau^{-1}\left(1 + \gamma^2 z_1 z_2\right)\mathbf{I} - iwA - (z_1 + z_2)\,\widehat{A}\right)^{-1} \mathbf{1}\, y_n, \\ &\widetilde{y}_{n+1} = y_n + \left(iwb + z\widehat{b}\right)^T \cdot \left(\tau^{-1}\left(1 + \gamma^2 z_1 z_2\right)\mathbf{I} - iwA - (z_1 + z_2)\,\widehat{A}\right)^{-1} \mathbf{1}\, y_n. \end{aligned}$$

The corresponding stability function is

$$\begin{aligned} &y_{n+1} = R(z_1, z_2, iw)\, y_n, \\ &R(z_1, z_2, iw) = 1 + \left(iwb + z\widehat{b}\right)^T \cdot \left(\tau^{-1}\left(1 + \gamma^2 z_1 z_2\right)\mathbf{I} - iwA - z\,\widehat{A}\right)^{-1} \mathbf{1}, \\ &R(\infty, \infty, iw) = 1. \end{aligned}$$

Two refinement steps lead to the stability function

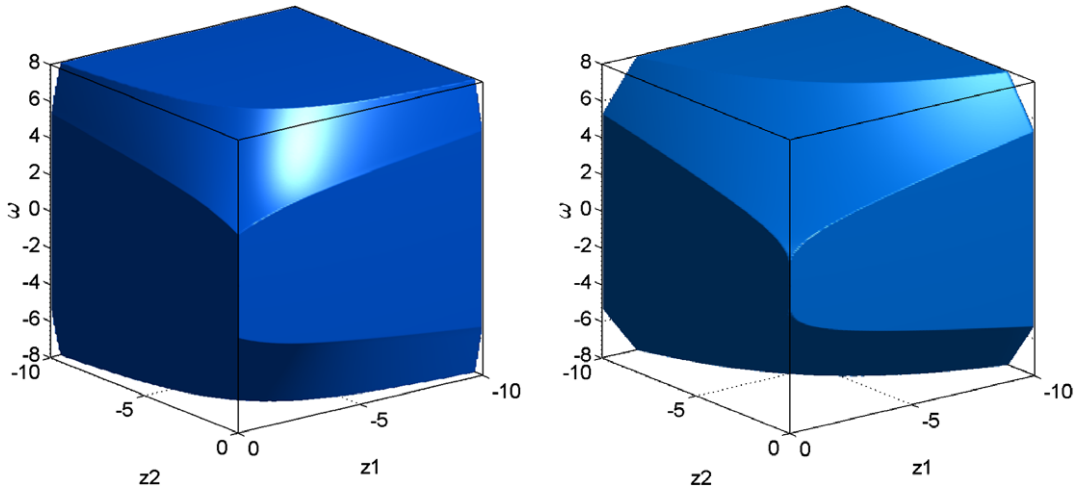
$$\begin{aligned} &y_{n+1} = R(z_1, z_2, iw)\, y_n, \\ &R(z_1, z_2, iw) = 1 + \left(iwb + z\widehat{b}\right)^T \cdot \left(\tau^{-2}\left(1 + \gamma^2 z_1 z_2\right)\mathbf{I} - iwA - z\,\widehat{A}\right)^{-1} \mathbf{1}, \\ &R(\infty, \infty, iw) = 1. \end{aligned}$$

We have that for very stiff components the scheme with refinement is weakly stable. The refinement does not improve the overall stability properties of the scheme, only the accuracy. The correction step of Calvo and Gerisch [16] leads to L-stable, first order methods for very stiff problems.

4. Numerical experiments

We perform numerical experiments with the following methods:

- **LIRK3(4):** the original LIRK methods of orders three and four, respectively, proposed by Calvo, de Frutos, and Novo [2]. The implicit parts are L-stable and stiffly accurate;
- **LIRK3(4)AMF:** the LIRK methods using AMF as (12);



(a) LIRK3 + AMF. (b) LIRK4 + AMF.

Fig. 1. Stability regions of LIRK with AMF in the volume $(z1, z2, \omega)$.

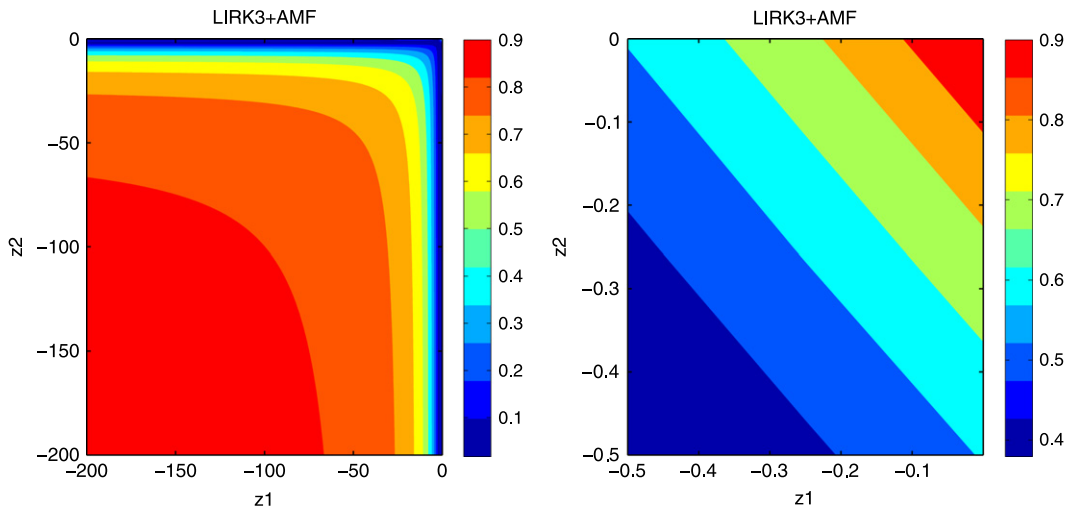


Fig. 2. Stability regions of LIRK3 with AMF in the plane $(z1, z2)$.

- **LIRK3(4)AMFR1:** the LIRK methods using AMF together with one iteration refinement;
- **LIRK3(4)AMFR2:** the LIRK methods using AMF together with two iterations refinement.

In all the experiments, the error is computed in the relative L_2 norm as follows

$$E = \frac{\|u - u_{\text{ref}}\|_2}{\|u_{\text{ref}}\|_2}, \tag{21}$$

where u is the numerical solution at the final time, and u_{ref} is the reference solution at the same point obtained using Matlab's `ode15s` solver with very tight tolerances (AbsTol = RelTol = 3×10^{-14}). We only look at the ODE error in this paper; in a real application one needs to account for both the spatial and temporal discretization errors.

4.1. An Allen–Cahn type problem

We use the PDE test problem of Allen–Cahn type from [16]:

$$u_t = \Delta u + u - u^3 + f, \tag{22}$$

where f is chosen to make the exact solution of the equation be

$$u(t, x, y) = e^t \sin(\pi x) \sin(\pi y).$$

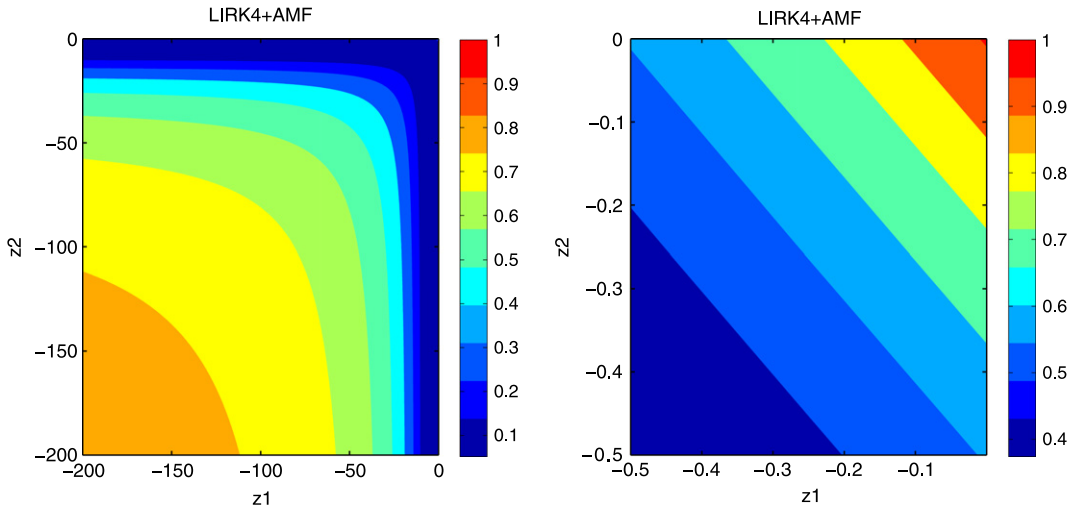


Fig. 3. Stability regions of LIRK4 with AMF in the plane $(z1, z2)$.

The spatial domain is $(x, y) \in [0, 1] \times [0, 1]$ and the time interval is $t \in [0, 1]$ (units). The initial conditions and Dirichlet boundary conditions are calculated from the exact solution.

The spatial discretization uses second order central finite differences of the Laplacian on a uniform grid of size $M \times M$

$$(x_i, y_i) = \left(\frac{i}{M+1}, \frac{j}{M+1}\right), \quad i, j = 1, \ldots, M. \tag{23}$$

In our tests we consider the case $M = 59$. The discrete solution elements $U_{ij}(t) \approx u(t, x_i, y_j)$ are ordered into a vector

$$z = (U_{11}, U_{12}, \ldots, U_{1M}, \ldots, U_{M1}, U_{M2}, \ldots, U_{MM})^T. \tag{24}$$

The resulting ODE system can then be written into the form (1) with the discrete diffusion term being the linear part. The largest magnitude of the eigenvalues of the Jacobian for the diffusion term is approximately 2.88×10^4. The discrete Laplacian operator $\mathbf{L} = \mathbf{L}_x + \mathbf{L}_y$ is split into two matrices corresponding to derivatives along x and y directions, respectively:

$$\mathbf{L}_x = \mathbf{D}_M \otimes \mathbf{I}_M, \qquad \mathbf{L}_y = \mathbf{I}_M \otimes \mathbf{D}_M, \qquad \mathbf{D}_M = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}, \tag{25}$$

where $\mathbf{I}_M$ is an identity matrix of dimension $M \times M$.

Fig. 4(a) plots the convergence results for all the methods tested. As expected, both LIRK3AMF and LIRK4AMF show second order, and give less accurate results for the same time step than the underlying LIRK methods. All the LIRK methods with AMF and refinement perform equally well as the original LIRK methods; LIRK3AMFR1 produces slightly better results, and the full order of the underlying LIRK methods has been recovered.

Fig. 4(b) shows the corresponding work-precision diagrams. LIRK methods with AMF are not very competitive in terms of efficiency. One refinement iteration improves LIRK3AMF and LIRK4AMF significantly. LIRK3AMFR1 and LIRK4AMFR1 are clearly the most efficient methods among the methods of the same order. A second iteration does not improve accuracy, but spends compute time, and makes LIRK3(4)AMFR2 schemes slightly less efficient.

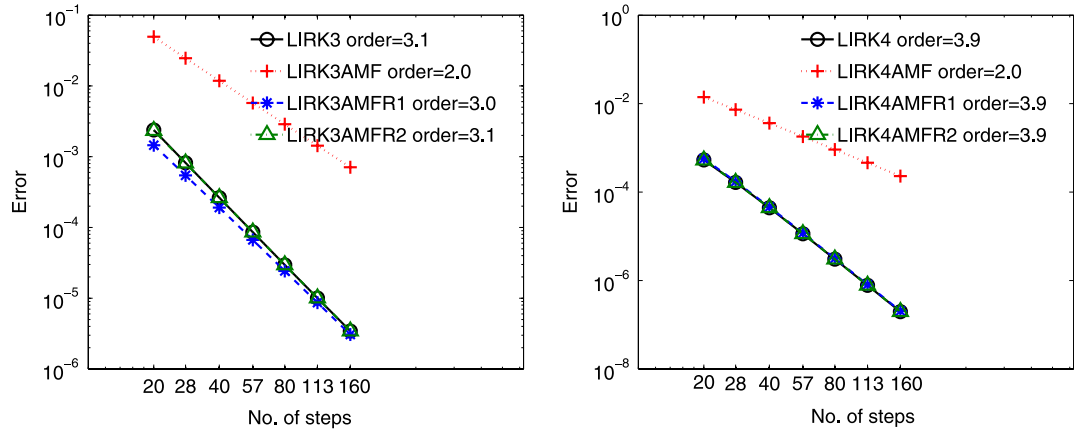
Calvo and Gerisch's approach [16] can achieve second order with AMF, and may only be attractive for low accuracy requirements. The approach with stage refinement proposed herein is competitive at all accuracy levels due to the recovery of the full order and the addition of the relatively cheap refinement procedure. It should be noted that the separation of $\mathbf{L}$ into $\mathbf{L}_x$ and $\mathbf{L}_y$ in (25) allows for a reduction of the large linear systems with $2M$ small ones of dimension $M \times M$, which can lead to considerable savings in the computational cost. The trick is not used with this example simply for the sake of enabling a fair comparison with the results in [16].

4.2. Brusselator problem

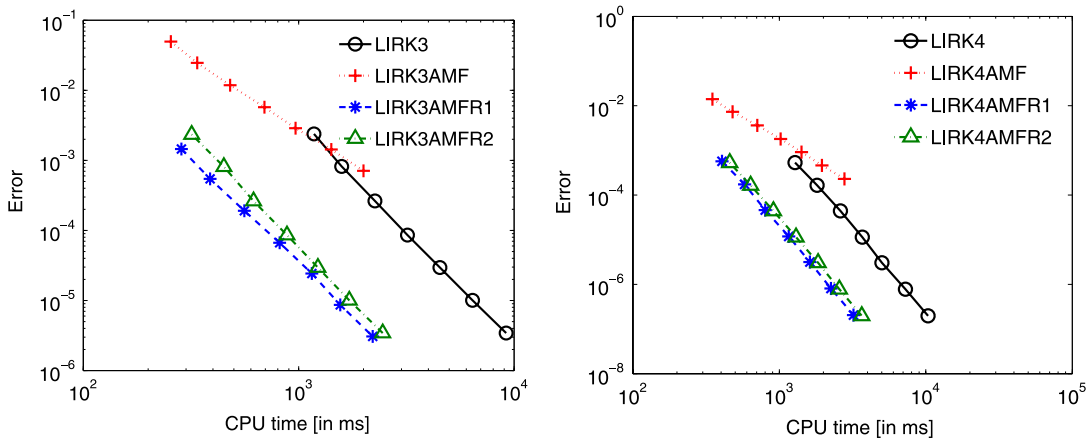
We next consider the two-dimensional Brusselator reaction–diffusion equation [19, Section IV.10]

$$u_t = 1 + u^2 v - (B+1)\, u + \alpha \Delta u, \tag{26a}$$

$$v_t = B\, u - u^2 v + \alpha \Delta v, \tag{26b}$$



(a) Temporal errors vs. number of steps.



(b) Temporal errors vs. CPU time.

Fig. 4. Results for the 2D Allen–Cahn type problem (22).

where $(x, y) \in [0, 1]^2$, $t \in [0, 1.5]$, with the Neumann boundary conditions

$$\frac{\partial u}{\partial \mathbf{n}} = 0, \qquad \frac{\partial v}{\partial \mathbf{n}} = 0.$$

The problem is discretized with a second order central finite difference scheme on a uniform mesh (23). The stiffness of this problem increases with the value of α and number of grid points M. We test LIRK + AMF methods with or without iterative corrections on two different cases.

Case 1. A nonstiff stiff case described by $\alpha = 0.001$, $B = 3$ and the initial conditions

$$u(x, y, 0) = 0.5 + y, \qquad v(x, y, 0) = 1 + 5x$$

with $M = 39$ grid points used in each dimension. This gives an ODE system of dimension $N = 3042$.

Case 2. A stiff case, in which the settings follow [19, Section IV.10] where $\alpha = 0.1$, $B = 3.4$, and the initial conditions are defined as

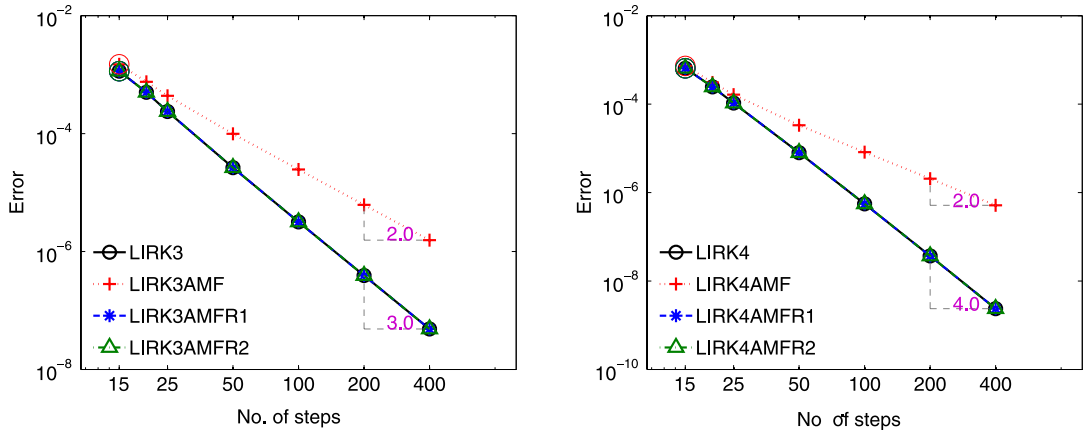
$$u(x, y, 0) = 22y(1 - y)^{3/2}, \qquad v(x, y, 0) = 22x(1 - x)^{3/2}.$$

We choose $M = 127$ for the three-way splitting test and $M = 199$ for the two-way splitting test so that the resulting ODE systems are of size $N = 31{,}752$ and $N = 79{,}202$ respectively. The time varying Jacobian for the reaction term used in the three-way splitting test makes the linearization challenging especially when the problem is very stiff. For a large M, e.g. $M = 150$, the error caused by the linearization leads to failure in solving the linear systems with direct methods after a few time steps. So we select a relatively smaller value of M for the three-way splitting test. The details on the splitting setup are given later in this section.

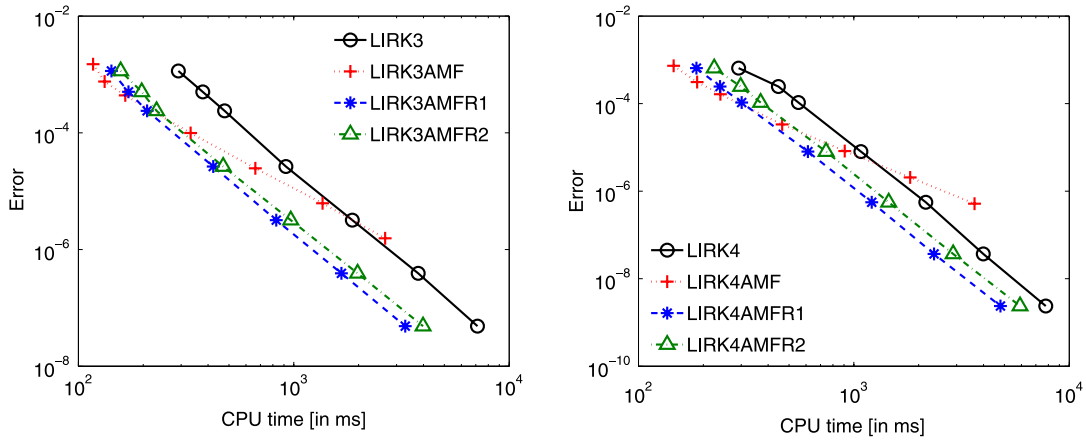
Table 1 shows the dominant eigenvalues for these test cases which shed light on the degree of stiffness. Note that the Jacobian matrix for the reaction term has complex eigenvalues which makes this test problem more challenging than the previous one.

Table 1
The dominant eigenvalues (largest in magnitude) of each component.

Case	$\mathbf{L}_x$	$\mathbf{L}_y$	$\mathbf{L}_x + \mathbf{L}_y$	$L_{rea}(t_0)$
1 ($M = 39$)	1.28×10^1	1.28×10^1	2.56×10^1	1.05×10^1
2 ($M = 127$)	6.55×10^3	6.55×10^3	1.31×10^4	2.01×10^1
3 ($M = 199$)	1.60×10^4	1.60×10^4	3.20×10^4	2.01×10^1



(a) Temporal error vs. number of steps.



(b) Temporal error vs. CPU time.

Fig. 5. Results for the 2D Brusselator system (26), Case 1, with $M = 39$. AMF is applied with a two-way splitting of the Jacobian. 15, 20, 25, 50, 100, 200, 400 equal steps are used for the time integration of the system on the interval [0, 1]. The left-most points (highlighted by a circle) on each curve indicate the maximal allowable time steps.

After spatial discretization the PDE is turned into a semi-linear ODE system of the form

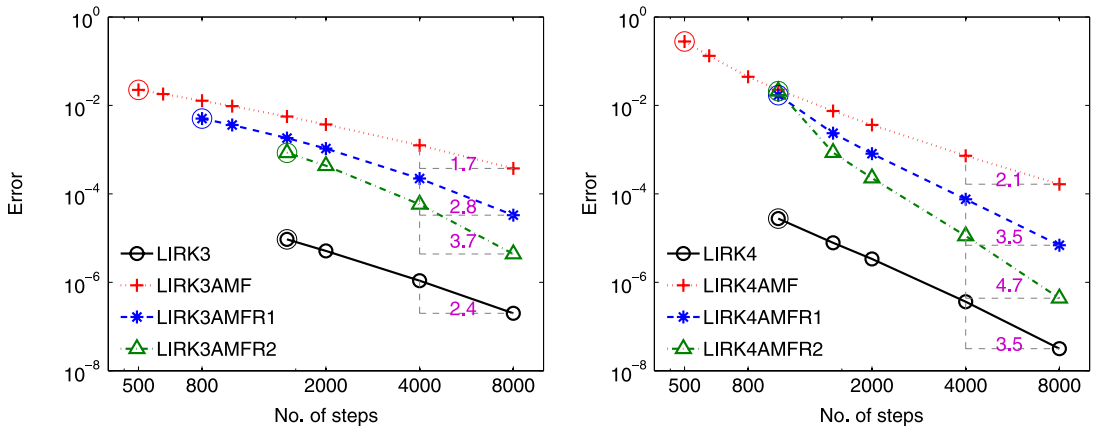
$$z' = \underbrace{\mathbf{L}_{dif}\, z}_{\text{diffusion}} + \underbrace{\mathbf{L}_{rea}(t)\, z}_{\text{reaction}} + R.$$

Here $z = [U^T\ V^T]^T$ with U and V defined as the vectors of the discrete solution elements for variables u and v, respectively (see also (24)), $\mathbf{L}_{dif}$ and $\mathbf{L}_{rea}(t)$ are the Jacobians of the diffusion term and reaction term, respectively, and R collects the rest of the terms such as boundary conditions. Note that $\mathbf{L}_{rea}(t)$ can be computed by

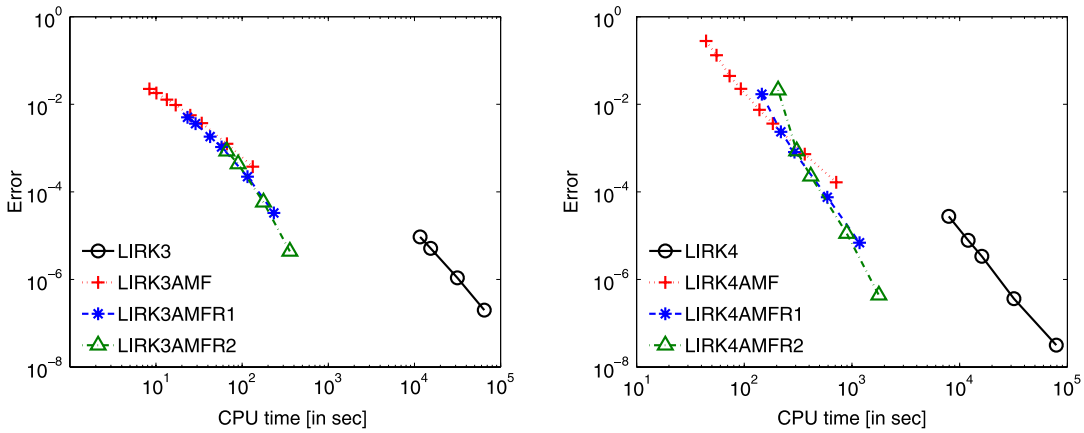
$$\mathbf{L}_{rea}(t) = \begin{pmatrix} \mathrm{diag}\,(-(B+1) + 2\,U \odot V) & \mathrm{diag}\,(U \odot U) \\ \mathrm{diag}\,(B - 2\,U \odot V) & \mathrm{diag}\,(-U \odot U) \end{pmatrix}, \tag{27}$$

and R does not need to be evaluated separately as shown in (9).

The diffusion term is stiff while the reaction term is nonstiff. We first apply LIRK methods with the diffusion treated implicitly and the other terms explicitly. Next we include the Jacobian of the reaction terms in the linear part and apply



(a) Temporal error vs. number of steps.



(b) Temporal error vs CPU time.

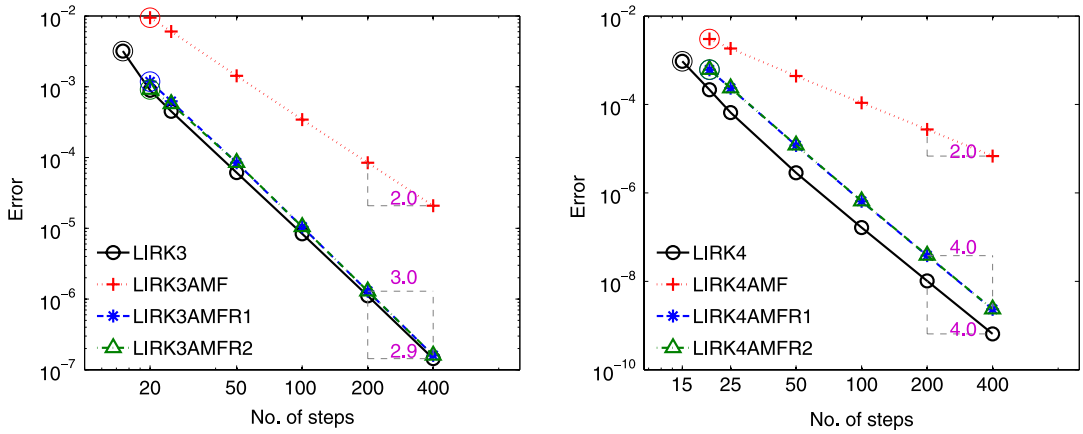
Fig. 6. Results for the 2D Brusselator system (26), Case 2, with $M = 199$. AMF is applied with a two-way splitting of the Jacobian. 400, 500, 600, 800, 1000, 1500, 2000, 4000, 8000 equal steps are used for the time integration of the system on the interval [0, 1]. The left-most points (highlighted by a circle) on each curve indicate the maximal allowable time steps.

a three-way splitting strategy. The LU decompositions are performed per time step using sparse Gaussian elimination in MATLAB.

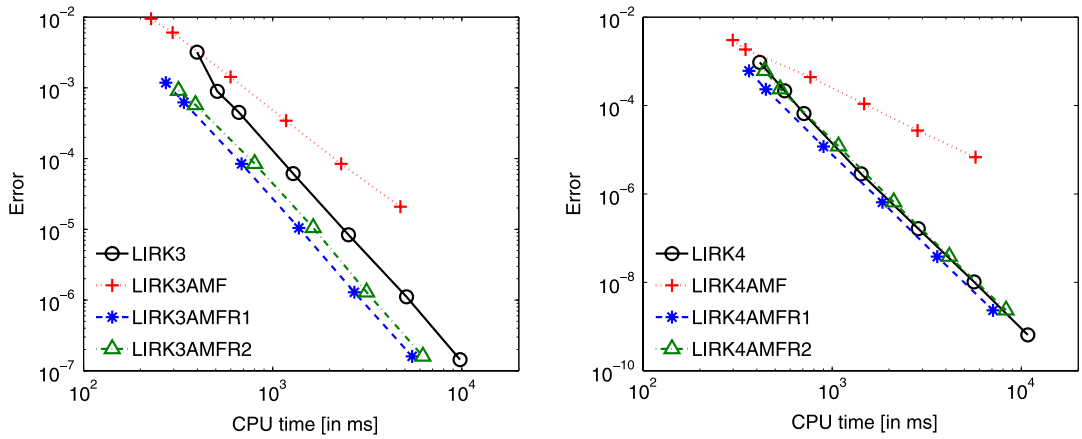
Two-way splitting. A directional splitting is applied to the diffusion term which is written as the sum of derivatives in the x-direction and y-direction, $\mathbf{L}_{\text{dif}} = \mathbf{L}_{\text{x}} + \mathbf{L}_{\text{y}}$. See (25) for the structure of $\mathbf{L}_{\text{x}}$ and $\mathbf{L}_{\text{y}}$. This splitting allows to reduce the linear algebra effort to solving $2M$ tridiagonal systems of dimension M, all of which share the same matrix $\mathbf{I} - h\gamma\,\mathbf{D}_{\mathbf{M}}$, and use reordered right-hand sides.

Fig. 5(a) shows the convergence plots of different methods for Case 1. Both LIRK3AMF and LIRK4AMF give second order. With one refinement iteration the results become as accurate as those of the original LIRK methods. The largest allowable time steps are almost the same for all methods, implying good stability properties of LIRK methods with AMF. Fig. 5(b) presents the corresponding work-precision diagrams. It can be seen that LIRK methods with one refinement iteration yield the best efficiency. To achieve the same accuracy level, LIRK3AMFR1 is about 2.2 times faster than LIRK3 and LIRK4AMFR1 is about 1.6 times faster than LIRK4.

Fig. 6 shows the convergence and work-precision diagrams for different methods for the large scale stiff case with $M = 199$. Generally LIRK methods with AMF and two refinement iterations give more accurate results than those with AMF and one refinement iteration. The refinement works well and recovers the theoretical orders of the corresponding LIRK methods. The errors for methods with two refinement iterations approach the LIRK results at a faster rate than methods with just one refinement iteration. This differs from the results of the first case, but is in line with the theoretical prediction. Another notable advantage of the AMF technique is the gain in term of stability. If no refinement is employed, the maximal time step size, indicated by the left-most point on each curve in Fig. 6(a), can be at least two times larger than that allowed by the original LIRK methods for both third-order and fourth-order schemes. However, the gain disappears or shrinks when the refinement procedure is added.



(a) Temporal error vs. number of steps.



(b) Temporal error vs. CPU time.

Fig. 7. Results for the 2D Brusselator system (26), Case 1, with $M = 39$. AMF is applied with a *three-way* splitting of the Jacobian. 15, 20, 25, 50, 100, 200, 400 equal steps are used for the time integration of the system on the interval [0, 1]. The left-most points (highlighted by a circle) on each curve indicate the maximal allowable time steps. The numbers inside the triangle give the convergence order.

In the efficiency comparison, LIRK methods with AMF and two refinement iterations are the most effective for solutions more accurate than approximately 10^{-4}. LIRK methods with AMF provide a good compromise between accuracy and speed since they can use a maximal allowable time step size that is at least two times larger than LIRK methods, and run significantly faster than LIRK methods.

Three-way splitting. We use a three-way splitting of the linear part

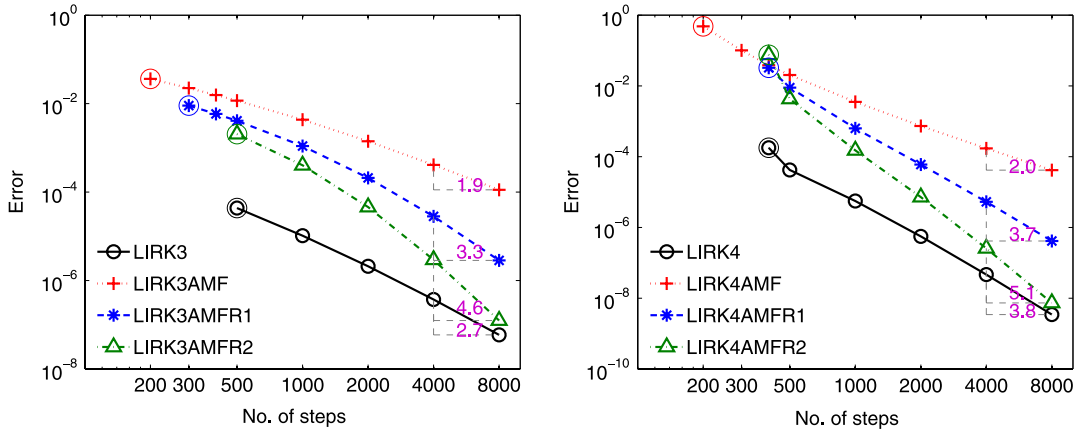
$$\mathbf{L} = \mathbf{L}_x + \mathbf{L}_y + \mathbf{L}_{rea}(t)$$

where the Jacobian of the reaction terms is treated implicitly. $\mathbf{L}_{rea}(t)$ contains four blocks, each of which is a diagonal matrix. It is updated at each time step.

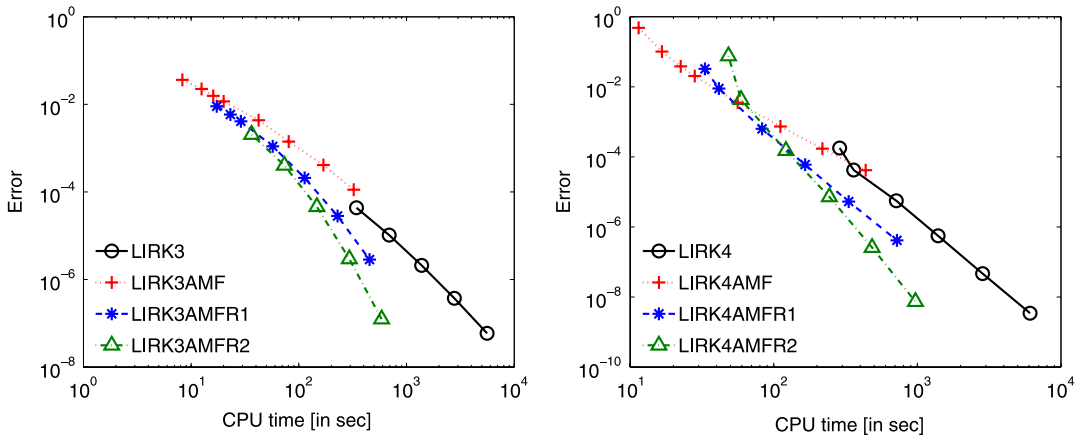
The linear system associated with $\mathbf{I} - h\gamma\, \mathbf{L}_{rea}$ can be reduced to M^2 smaller systems of dimension two which can be solved separately at each grid point. We choose to solve the large sparse system directly as an explicit decoupling does not lead to a clear performance gain for a serial implementation.

The results are shown in Figs. 7, and 8. For Case 1, the refinement procedure can successfully improve the order from 2 to the theoretical order. AMF without refinement is not competitive in terms of efficiency, but AMF with refinement yields some performance gain for third-order schemes and comparable results with LIRK methods for fourth-order schemes. This is due to the fact that the system is relatively small and additional cost is brought in to solve the linear system associated with $\mathbf{L}_{rea}(t)$.

For the stiff case, the Jacobian for the implicit part $\mathbf{L}$ makes the linear system difficult to solve with direct methods. One LU decomposition of the system may take over 5000 s. To improve the performance of LIRK methods, we make use of the reordering algorithm `symamd` in MATLAB before solving the linear systems. The reordering can also help reduce the bandwidth of the sparse matrices as similar to purpose of the splitting schemes we used.



(a) Temporal error vs. number of steps.



(b) Temporal error vs. CPU time.

Fig. 8. Results for the 2D Brusselator system (26), Case 2, with $M = 127$. AMF is applied with a three-way splitting of the Jacobian. 100, 200, 300, 400, 500, 1000, 2000, 4000, 8000 equal steps are used for the time integration of the system on the interval [0, 1]. The left-most points (highlighted by a circle) on each curve indicate the maximal allowable time steps.

The convergence orders for the stiff case are very close to the two-way splitting test results. In terms of efficiency, there is still considerable performance gain for AMF with refinement, especially for AMF with two refinement iterations. The savings in CPU time may mainly come from reducing the big system into multiple small systems, which is another advantage of the application of AMF.

5. Conclusions

We have applied approximate matrix factorization to high order linearly implicit Runge–Kutta methods for solving semi-linear systems of differential equations. The factorization (splitting) error brought by AMF leads to severe order degradation, especially for high order Runge–Kutta methods. The existing approach to recover second order is based on correction applied to the next step solution [16]. In this work the full order of the underlying methods is recovered by correcting stage values via a refinement procedure based on the idea of simplified Newton iterations.

We have performed error analysis for the linear system solutions with AMF, and investigated how this errors affect the next step solution. In the non-stiff and mildly stiff case the full order of the underlying method can be recovered using a fixed, small number of refinement iterations. In the very stiff case the number of iterations can be large since the convergence can deteriorate with increasing stiffness. A stability analysis reveals that the stage refinement procedure does not improve the overall stability of the LIRK + AMF method. When AMF is used the resulting schemes are only weakly stable for very stiff problems. Consequently, this application of AMF is attractive for mildly stiff problems, but may not work well for very stiff systems.

Numerical experiments on a variety of test problems of different sizes and different degrees of stiffness validate the theoretical findings on the accuracy and stability of high order linearly implicit Runge–Kutta methods when AMF is used. The results also show that the proposed approach can improve the efficiency of high order linearly implicit Runge–Kutta

methods significantly and thus is attractive for solving large scale mildly stiff systems such as diffusion–reaction equations. Furthermore, our tests on the three-way splitting demonstrate that our methods can also efficiently deal with problems where stiffness comes from both diffusion and reaction terms. Though we considered LIRK schemes up to order four, the general framework developed herein can be applied to higher order LIRK methods, and could be extended to study the use of AMF with implicit–explicit multistep methods and implicit–explicit general linear methods. Future work will look at adaptive step sizes, which may allow larger time steps for the full method.

Acknowledgments

This work has been supported in part by National Science Foundation through awards NSF CMMI–1130667, NSF CCF–1218454, NSF CCF–0916493, NSF DMS–1419003, AFOSR FA9550–12–1–0293–DEF, AFOSR 12-2640-06, and by the Computational Science Laboratory at Virginia Tech.

Appendix. LIRK methods

The coefficients of the LIRK3 method [2]:

$$\begin{array}{c|ccc} 0 & 0 & & \\ \gamma & 0 & \gamma & \\ \frac{1+\gamma}{2} & 0 & \frac{1-\gamma}{2} & \gamma \\ 1 & 0 & b_2 & b_3 \quad \gamma \\ \hline & 0 & b_2 & b_3 \quad \gamma \end{array} \qquad \begin{array}{c|cccc} 0 & 0 & & & \\ \gamma & \gamma & 0 & & \\ \frac{1+\gamma}{2} & \frac{1-\gamma}{2} - a_{32} & a_{32} & 0 & \\ 1 & 0 & 1 - a_{43} & a_{43} & 0 \\ \hline & 0 & b_2 & b_3 & \gamma \end{array},$$

where $b_2 = -\frac{3\gamma^2}{2} + 4\gamma - \frac{1}{4}$ and $b_3 = \frac{3\gamma^2}{2} - 5\gamma + \frac{5}{4}$. And the choice for the free parameter is $\gamma = 0.435866521508459$ and $a_{32} = 0.35$.

The coefficients of the LIRK4 method [2]:

$$\begin{array}{c|cccccc} 0 & & & & & & \\ \frac{1}{4} & 0 & \frac{1}{4} & & & & \\ \frac{3}{4} & 0 & \frac{1}{2} & \frac{1}{4} & & & \\ \frac{11}{20} & 0 & \frac{17}{50} & -\frac{1}{25} & \frac{1}{4} & & \\ \frac{1}{2} & 0 & \frac{371}{1360} & -\frac{137}{2720} & \frac{15}{544} & \frac{1}{4} & \\ 1 & 0 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4} \\ \hline & 0 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4} \end{array} \qquad \begin{array}{c|cccccc} 0 & 0 & & & & & \\ \frac{1}{4} & \frac{1}{4} & 0 & & & & \\ \frac{3}{4} & -\frac{1}{4} & 1 & 0 & & & \\ \frac{11}{20} & -\frac{13}{100} & \frac{43}{75} & \frac{8}{75} & 0 & & \\ \frac{1}{2} & -\frac{6}{85} & \frac{42}{85} & \frac{179}{1360} & -\frac{15}{272} & 0 & \\ 1 & 0 & \frac{79}{24} & -\frac{5}{8} & \frac{25}{2} & -\frac{85}{6} & 0 \\ \hline & 0 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4} \end{array}.$$

References

[1] I. Grooms, K. Julien, Linearly implicit methods for nonlinear PDEs with linear dispersion and dissipation, J. Comput. Phys. 230 (9) (2011) 3630–3650.
[2] M.P. Calvo, J. de Frutos, J. Novo, Linearly implicit Runge–Kutta methods for advection–reaction–diffusion equations, Appl. Numer. Math. 37 (4) (2001) 535–549.
[3] G. Akrivis, O. Karakashian, F. Karakatsani, Linearly implicit methods for nonlinear evolution equations, Numer. Math. 94 (2003) 403–418.
[4] G. Akrivis, M. Crouzeix, Linearly implicit methods for nonlinear parabolic equations, Math. Comp. 73 (246) (2004) 613–635.
[5] R. Weiner, B.A. Schmitt, H. Podhaisky, ROWMAP—a ROW-code with Krylov techniques for large stiff ODEs, Appl. Numer. Math. 25 (1997) 303–319.
[6] J. Douglas Jr., Alternating direction methods for three space variables, Numer. Math. 4 (1) (1962) 41–63.
[7] D. Peaceman, H. Rachford Jr., The numerical solution of parabolic and elliptic differential equations, J. Soc. Ind. Appl. Math. 3 (1) (1955) 28–41.
[8] A. Sandu, Numerical aspects of air quality modeling (Ph.D. thesis), University of Iowa, 1997.
[9] P. van der Houwen, B. Sommeijer, Approximate factorization for time-dependent partial differential equations, J. Comput. Appl. Math. 128 (1–2) (2001) 447–466.
[10] S. González-Pinto, D. Hernández-Abreu, S. Pérez-Rodríguez, Rosenbrock-type methods with inexact AMF for the time integration of advection–diffusion–reaction PDEs, J. Comput. Appl. Math. 262 (2014) 304–321.
[11] S. Beck, S. González-Pinto, S. Pérez-Rodríguez, R. Weiner, A comparison of AMF- and Krylov-methods in MATLAB for large stiff ODE systems, J. Comput. Appl. Math. 262 (2014) 292–303. selected Papers from NUMDIFF-13.
[12] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer, 1993.
[13] S. Beck, R. Weiner, H. Podhaisky, B. Schmitt, Implicit peer methods for large stiff ODE systems, J. Appl. Math. Comput. 38 (1–2) (2012) 389–406.
[14] M. Berzins, J. Ware, Solving convection and convection–reaction problems using the method of lines, Appl. Numer. Math. 20 (1–2) (1996) 83–99. method of Lines for Time-Dependent Problems.
[15] I. Ahmad, M. Berzins, An algorithm for ODEs from atmospheric dispersion problems, Appl. Numer. Math. 25 (2–3) (1997) 137–149. special Issue on Time Integration.
[16] M.P. Calvo, A. Gerisch, Linearly implicit Runge–Kutta methods and approximate matrix factorization, Appl. Numer. Math. 53 (2) (2005) 183–200.
[17] U. Ascher, S. Ruuth, R. Spiteri, Implicit–explicit Runge–Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (1997) 151–167.
[18] L. Pareschi, G. Russo, Implicit–explicit Runge–Kutta schemes for stiff systems of differential equations, in: Recent Trends in Numerical Analysis, Nova Science Publishers, Inc., 2000, pp. 269–288.
[19] E. Hairer, S. Norsett, G. Wanner, Solving Ordinary Differential Equations I. Nonstiff Problems, Springer-Verlag, Berlin, 1993.