Martin–Luther–Universität
Halle–Wittenberg
Institut für Mathematik

High-order finite element – linearly implicit two-step peer methods for time-dependent PDEs

A. Gerisch, J. Lang, H. Podhaisky, and R. Weiner

Report No. 13 (2006)

Editors:
Professors of the Institute for Mathematics, Martin-Luther-University Halle-Wittenberg.

Electronic version: see `http://www.mathematik.uni-halle.de/reports`

High-order finite element – linearly implicit two-step peer methods for time-dependent PDEs

A. Gerisch, J. Lang, H. Podhaisky, and R. Weiner

Report No. 13 (2006)

Dr. Alf Gerisch
Dr. Helmut Podhaisky
Prof. Dr. Rüdiger Weiner
Naturwissenschaftliche Fakultät III
Institut für Mathematik
Arbeitsgruppe Numerische Mathematik
Martin–Luther–Universität Halle–Wittenberg
Theodor-Lieser-Str. 5
D-06120 Halle/Saale, Germany
Email: {gerisch,podhaisky,weiner,}@mathematik.uni-halle.de

Prof. Dr. Jens Lang
Technische Universität Darmstadt
FB Mathematik, AG 12
Schloßgartenstr. 7
D-64289 Darmstadt, Germany
Email: lang@mathematik.tu-darmstadt.de

High-order finite element – linearly implicit two-step peer methods for time-dependent PDEs*

A. Gerisch†,[1] J. Lang,[2] H. Podhaisky,[1] and R. Weiner[1]

[1] Martin-Luther-Universität Halle-Wittenberg
Institut für Mathematik
Postfach
D-06099 Halle (Saale), Germany
{gerisch,podhaisky,weiner}@mathematik.uni-halle.de

[2] Technische Universität Darmstadt
FB Mathematik, AG 12
Schloßgartenstr. 7
D-64289 Darmstadt, Germany
lang@mathematik.tu-darmstadt.de

Abstract

Linearly-implicit two-step peer methods are successfully applied in the numerical solution of ordinary differential and differential-algebraic equations. One of their strengths is that even high-order methods do not show order reduction in computations for stiff problems. With this property, peer methods commend themselves as time-stepping schemes in finite element calculations for time-dependent partial differential equations (PDEs).

We have included a class of linearly-implicit two-step peer methods in the finite element software KARDOS. There PDEs are solved following the Rothe method, i.e. first discretised in time, leading to linear elliptic problems in each stage of the peer method. We describe the construction of the methods and how they fit into the finite element framework. We also discuss the starting procedure of the two-step scheme and questions of local temporal error control.

The implementation is tested for two-step peer methods of orders three to five on a selection of PDE test problems on fixed spatial grids. No order reduction is observed and the two-step methods are more efficient, at least competitive, in comparison with the linearly implicit one-step methods provided in KARDOS.

Mathematics Subject Classification 2000: 65M20, 65M60, 65L06
Keywords: Rosenbrock method · Peer method · Two-step method · Time-dependent PDE · Finite element software · KARDOS

1 Introduction

KARDOS [2] is a finite element software for the numerical solution of time-dependent nonlinear systems of partial differential and partial differential algebraic equations (PDEs and PDAEs). KARDOS implements the adaptive Rothe method with linearly implicit one-step methods for time stepping and an h-adaptive finite element (FE) method for the solution of the resulting time-independent stage problems.

Many difficult problems arising in areas such as the life sciences (pattern formation, regional hyperthermia of tumours [7]), diffusion processes [8], and propagation of flame fronts [3] can be solved efficiently with KARDOS. This efficiency is achieved by using adaptivity in space but also by the use

*This manuscript appears as *Report on Numerical Mathematics No. 06-13*, Martin-Luther-Universität Halle-Wittenberg, Germany, November 7, 2006. Online available at `http://www.mathematik.uni-halle.de/reports/`. Submitted for publication.

†corresponding author

of higher order time stepping methods such as `ros3p` [9], `rodas` [4], and `rodasp` [18, 19]. However, those time integration methods do not always show the expected classical order of convergence but smaller, often fractional orders of convergence in PDE applications. This phenomenon is known as *order reduction* and can be understood and analysed in the context of abstract ordinary differential equations (ODEs) in Hilbert spaces, see e.g. [11, 10]. Additional order conditions can be derived to ensure that no order reduction takes places but to find methods satisfying these conditions becomes increasingly difficult with increasing order of convergence. Methods of that kind are `ros3p` (order 3 for general nonlinear parabolic problems) and `rodasp` (order 4 for linear parabolic problems). A one-step linearly-implicit method of order 4 or higher for general nonlinear parabolic problems is not available to the best of our knowledge.

The aim of the present paper is to introduce higher order time stepping schemes in KARDOS which do not suffer from order reduction. This is achieved by going from one-step to two-step methods. As a particular class so-called *two-step peer methods* [16, 20, 14, 17] commend themselves for that task. These methods have shown no order reduction in applications to stiff ODE systems and we will use a slightly modified class of such methods in KARDOS. The attribute *peer* refers to the fact that all approximations to the exact solution computed within one time step of the method have the same order of accuracy. This is in contrast to the distinction between stage values (usually having low order) and the finally computed approximation (having a higher order) at the end of each time step of a one-step method.

The remainder of the paper unfolds as follows. Section 2 introduces the class of two-step peer methods which we find suitable for inclusion into the KARDOS system. Here we still consider the classical ODE setting and give order and stability results of the methods. We also comment on some of the guiding design principles for the determination of the free method coefficients. In Section 3 we then describe the inclusion of these methods into the KARDOS system. Here we consider the PDE to be solved as an abstract ODE and then apply the peer method. We comment on the starting procedure of the two-step schemes and give details of the time step control. In this paper we concentrate on the case of a fixed spatial grid; questions of spatial adaptivity and its interplay with the time step selection will be covered in forthcoming work. We demonstrate the efficiency and robustness of the new time stepping schemes in KARDOS by presenting selected numerical experiments in Section 4. Finally, Section 5 summarises the results, presents conclusions drawn from the numerical experiments and gives an outlook on ongoing work.

2 Peer methods for ODEs

We consider the numerical solution of initial value problems for systems of ODEs, for simplicity in autonomous form,

$$\dot{y}(t) = f(y(t)), \quad y(t_0) = y_0 \in \mathbb{R}^n \tag{1}$$

by means of s-stage two-step peer methods.

The time step $m \geq 1$ of size $\tau_m > 0$ of an s-stage *diagonally implicit two-step peer method* computes approximations $Y_{mi} \in \mathbb{R}^n$, $i = 1, \ldots, s$, as approximations to the exact solution at time $t = t_{mi}$, i.e. $Y_{mi} \approx y(t_{mi})$ from the scheme

$$Y_{mi} = \tau_m \gamma f(Y_{mi}) + w_i, \quad i = 1, \ldots, s, \tag{2a}$$

$$w_i = \sum_{j=1}^{i-1} a_{ij} \tau_m f(Y_{mj}) + \sum_{j=1}^{s} u_{ij}(\sigma_m) Y_{m-1,j}, \tag{2b}$$

where each vector w_i is computed from already computed quantities. Here, $A = (a_{ij})$ and $U = (u_{ij})$ are $s \times s$ real coefficient matrices: matrix A is lower triangular with constant entries $a_{ii} = \gamma > 0$ and matrix U can be a full matrix. It will turn out that U must depend on the step size ratio $\sigma_m := \tau_m/\tau_{m-1}$ of two successive time steps for ensuring the order of the method for variable step sizes. Furthermore, $c \in \mathbb{R}^s$ with $c_s = 1$ denotes the vector of pairwise distinct abscissae of the method. Note that we do not pose any further restrictions so far on the values of the c_i, in particular they are not confined to be in the interval $[0, 1]$. The time points t_{mi} are defined by $t_{mi} := t_m + c_i\tau_m$ for $m = 1, 2, \dots, i = 1, \dots, s$, where $t_m := t_{m-1,s}$ for $m > 1$. In order to perform step $m = 1$ of the method, we must provide the positive time step sizes τ_0 and τ_1, the time point t_1 and the s approximations $Y_{0i} \approx y(t_1 - \tau_0 + c_i\tau_0)$. How this data can be provided will be subject of Section 2.3. We note however already here that $t_0 \neq t_1 - \tau_0$ in general. For subsequent use, we collect all approximations Y_{mi} of one time step into a matrix

$$Y_m := (Y_{m1}, Y_{m2}, \dots, Y_{ms}) \in \mathbb{R}^{n,s}\,, \quad m \geq 0\,.$$

Remark 1 *The methods* (2) *fall into the class of General Linear Methods (GLMs) [1]. They form the special case of GLMs where all s internal stages are directly passed on to the next time step. Hence the tableaux of coefficients of the corresponding GLM has the form*

$$\left[\begin{array}{c|c} A & U \\ \hline B & V \end{array}\right] = \left[\begin{array}{c|c} A & U(\sigma_m) \\ \hline A & U(\sigma_m) \end{array}\right]$$

with A and $U(\sigma_m)$ from above.

It is easily seen that in method (2) the vectors Y_{mi} and $f(Y_{mi})$ can be used interchangeably. In fact, Eq. (2a) can be rearranged to

$$\tau_m f(Y_{mi}) = \frac{1}{\gamma}(Y_{mi} - w_i)\,, \quad i = 1, \dots, s\,.$$

Inserting this into Eq. (2b) results in an equivalent formulation of method (2) given by

$$Y_{mi} = \tau_m\gamma f(Y_{mi}) + w_i\,, \quad i = 1, \dots, s\,, \tag{3a}$$

$$w_i = \sum_{j=1}^{i-1} \frac{1}{\gamma} a_{ij}(Y_{mj} - w_j) + \sum_{j=1}^{s} u_{ij}(\sigma_m) Y_{m-1,j}\,, \tag{3b}$$

where the vectors w_i are now defined recursively.

Remark 2 *We observe that we can solve the recursion* (3b)*. For this collect all w_i column-wise in matrix $W \in \mathbb{R}^{n,s}$. Then* (3b) *can be written compactly by*

$$W^{\mathsf{T}} = \frac{1}{\gamma}(A - \gamma I)(Y_m - W)^{\mathsf{T}} + U(\sigma_m)Y_{m-1}^{\mathsf{T}}\,.$$

Solving for W results in

$$W^{\mathsf{T}} = (I - \gamma A^{-1})Y_m^{\mathsf{T}} + \gamma A^{-1}U(\sigma_m)Y_{m-1}^{\mathsf{T}}\,. \tag{4}$$

We define the matrices $\tilde{A} := I - \gamma A^{-1}$ and $\tilde{U}(\sigma_m) := \gamma A^{-1}U(\sigma_m)$ and note that $\tilde{A}$ is strictly lower triangular. Then we have

$$w_i = \sum_{j=1}^{i-1} \tilde{a}_{ij} Y_{mj} + \sum_{j=1}^{s} \tilde{u}_{ij}(\sigma_m) Y_{m-1,j}.$$

Eq. (3a) represents a nonlinear system of equations for Y_{mi}. Provided a good predictor Y^0_{mi} is available, one Newton step is often sufficient for its accurate solution. In Section 2.1 we provide appropriate predictors. Accordingly, from now on, we *define* Y_{mi} from

$$\left(\frac{I}{\tau_m\gamma} - T_m\right)(Y_{mi} - Y^0_{mi}) = f(Y^0_{mi}) + \frac{1}{\tau_m\gamma}\left(w_i - Y^0_{mi}\right) \tag{5a}$$

$$Y^0_{mi} = \sum_{j=1}^{i-1} \frac{1}{\gamma} a^0_{ij}(Y_{mj} - w_j) + \sum_{j=1}^{s} u^0_{ij}(\sigma_m) Y_{m-1,j} \tag{5b}$$

where $A^0 = (a^0_{ij})$ and $U^0 = (u^0_{ij})$ are additional $s \times s$ real coefficient matrices: A^0 is strictly lower triangular and $U^0 = U^0(\sigma)$ can be a full matrix which will depend again on the step size ratio. The matrix $T_m \in \mathbb{R}^{n,n}$ is constant for each time step and is an approximation to the Jacobian of f at $Y_{m-1,s}$, i.e.

$$T_m \approx \frac{\partial\, f(Y_{m-1,s})}{\partial\, y}\,.$$

The equations (5) replace Eq. (3a) leading to the *linearly implicit two-step peer methods* considered in this work

$$w_i = \sum_{j=1}^{i-1} \frac{1}{\gamma} a_{ij}(Y_{mj} - w_j) + \sum_{j=1}^{s} u_{ij}(\sigma_m) Y_{m-1,j}\,, \tag{6a}$$

$$Y^0_{mi} = \sum_{j=1}^{i-1} \frac{1}{\gamma} a^0_{ij}(Y_{mj} - w_j) + \sum_{j=1}^{s} u^0_{ij}(\sigma_m) Y_{m-1,j}\,, \tag{6b}$$

$$\left(\frac{I}{\tau_m\gamma} - T_m\right)(Y_{mi} - Y^0_{mi}) = f(Y^0_{mi}) + \frac{1}{\tau_m\gamma}\left(w_i - Y^0_{mi}\right)\,. \tag{6c}$$

Remark 3 *The method corresponding to method* (6) *for non-autonomous problems is obtained by replacing* $f(Y^0_{mi})$ *by* $f(t_{mi}, Y^0_{mi})$.

Remark 4 *The peer method* (6) *is derived for the numerical solution of initial value problems of ODEs* (1). *The corresponding method for the numerical solution of DAEs*

$$P\dot{y}(t) = f(y(t))\,, \quad y(t_0) = y_0 \in \mathbb{R}^n\,, \quad P \in \mathbb{R}^{n,n} \tag{7}$$

with a constant, possibly singular matrix P *is given by* (6a) *and* (6b) *for the computation of* w_i *and* Y^0_{mi}, *respectively, and* (6c) *replaced by*

$$\left(\frac{P}{\tau_m\gamma} - T_m\right)(Y_{mi} - Y^0_{mi}) = f(Y^0_{mi}) + \frac{P}{\tau_m\gamma}\left(w_i - Y^0_{mi}\right)\,. \tag{8}$$

This scheme is derived by considering the methods for singularly perturbed problems and then letting the perturbation parameter tend to zero. For details we refer to [13].

2.1 Construction of methods: order conditions and linear stability

In this subsection we briefly review the derivation of suitable parameter sets for methods of the form (6), for details see [14, 20].

We look first at the diagonally implicit methods Eq. (2), or equivalently (3), and ensure order $p = s - 1$ for those. A major feature of peer methods is their uniform order of consistency for all quantities

computed, i.e. $Y_{mi} = y(t_{mi}) + \mathcal{O}(\tau_m^{p+1})$. The abscissae c_i are chosen in advance to be stretched Chebychev nodes

$$c_i := -\frac{\cos((i-\frac{1}{2})\frac{\pi}{s})}{\cos(\frac{\pi}{2s})}\,, \quad i = 1, 2, \ldots, s$$

lying in the interval $[-1, 1]$. To ensure the local order $p = s - 1$, Eq. (2) is expanded in Taylor series yielding a linear relation between A and U which is regular and can be solved for U for all step size ratios $\sigma > 0$, i.e. $U = U(\sigma, A)$. It remains to find A and stability considerations are first exploited for that. We apply the peer method in the form (3) with constant step size τ, i.e. $\sigma = 1$, to the scalar linear test equation $y' = \lambda y$ with $\lambda \in \mathbb{C}$. This yields, using Eq. (4), the relation

$$Y_m^{\mathsf{T}} = M(z) Y_{m-1}^{\mathsf{T}}\,, \quad \text{with} \quad M(z) = (I - zA)^{-1}U\,, \quad z := \lambda\tau\,.$$

Here, $M(z)$ is the stability matrix of the method. For optimal zero stability we require that $M(0) = U$ has one eigenvalue 1 and all others equal to 0, i.e. vanishing parasitic roots. Following [14], we choose A such that U has this spectrum for all values of the step size ratio $\sigma > 0$. This leaves $\gamma = a_{ii}$ as a free parameter. Furthermore, observe that the stability matrix of the method vanishes at infinity, that is $M(\infty) = 0$. Hence we have that A(α)-stability is equivalent to L(α)-stability for the methods under consideration. A crucial problem is to find methods which are L-stable or at least L(α)-stable with a large angle α, as this is not automatically guaranteed. This angle is essentially determined by the value of γ, the remaining free parameter. We use the γ-values given in Table 1 (lines marked with $*$) in [14] for number of stages equal to $s = 4, 5$ and 6. Besides a large angle α, these values also ensure that the method has order $p = s$ for constant time step sizes and has coefficients of small magnitude.

To fully define the linearly implicit two-step peer methods (6), we need to construct the predictor matrices A^0 and U^0. This is much simpler than the derivation of A and U outlined above: we impose similar order condition for order $p = s - 1$ for the predictors Y_{mi}^0 and use all remaining degrees of freedom to minimise the magnitude of the coefficients, cf. [20]. Since each predictor in a method (6) now has order $p = s-1$ and the underlying diagonally implicit method (2) also has order $p = s-1$, it is clear that with the parameters chosen the method (6) has order $p = s-1$. Furthermore, since methods (6) and (2) coincide for linear problems, both methods also have the same linear stability properties.

The numerical results presented in Sec. 4 are obtained using methods (6) with number of stages equal to $s = 4, 5$ and 6. These are referred to as `peer4`, `peer5` and `peer6` in this work. They have order $p = s - 1$ for variable step size sequences and even order $p = s$ for constant time steps. The angles of L(α)-stability are $89.9°$, $89.3°$, and $85.4°$, respectively.

2.2 Computation of a solution of order $\tilde{p} = s - 2$

All methods considered are constructed such that they have order $p = s - 1$ for variable step sizes. In order to control the time step size we compute additional approximations $\tilde{Y}_{ms}$ at the time points t_m of order $p = s - 2$ as linear combinations of the $Y_{mi}, i = 1, \ldots, s - 1$. The local error in step m can then be approximated as

$$err_m := Y_{m,s} - \tilde{Y}_{m,s} = Y_{m,s} - \sum_{i=1}^{s-1} \alpha_i Y_{m,i}\,,$$

with coefficients α_i to be determined. Observe that by Taylor expansion

$$\sum_{i=1}^{s-1} \alpha_i Y_{m,i} = \sum_{i=1}^{s-1} \alpha_i y(t_m + c_i\tau_m) + \mathcal{O}\left(\tau_m^s\right) = \sum_{i=1}^{s-1} \alpha_i \sum_{l=0}^{s-2} y^{(l)}(t_m)\frac{c_i^l \tau_m^l}{l!} + \mathcal{O}\left(\tau_m^{s-1}\right)\,.$$

On the other hand,

$$y(t_m + \tau_m) = \sum_{l=0}^{s-2} y^{(l)}(t_m)\frac{\tau_m^l}{l!} + \mathcal{O}\left(\tau_m^{s-1}\right) .$$

Let $V_{0,s-1} = (c_i^l)$, $i = 1, ..., s-1$, $l = 0, ..., s-2$ be the Vandermonde matrix for $c_1, \ldots, c_{s-1}$. Then, by equating equal powers of τ_m for $l = 0, \ldots, s-2$, we obtain the coefficients α_i from

$$\alpha := V_{0,s-1}^{-\mathsf{T}} \mathbb{1} \in \mathbb{R}^{s-1} .$$

2.3 Starting procedure

For the execution of step $m = 1$ of a two-step method (6) we need to know the s approximations $Y_{0i} \approx y(t_1 - \tau_0 + c_i\tau_0)$, the time point t_1 and the two step sizes τ_0 and τ_1. We remark that $t_1 - \tau_0$ can be, and in general is, different from the initial time t_0 of the initial value problem.

In order to obtain the required approximations, we perform one time step of size $\tau_{\text{osm}} > 0$ of a suitable one-step method with continuous output using the initial data y_0 of the problem at time $t = t_0$. This gives access to a numerical solution $\tilde{y}(t)$ in the interval $[t_0, t_0 + \tau_{\text{osm}}]$. The accuracy of this numerical solution can be controlled by standard time step control or by choosing τ_{osm} sufficiently small. Denote by $c^- < c^+$ the minimum and maximum value of the abscissae values c_i, respectively. We now require that the smallest resp. largest time point at which an approximation for the solution $y(t)$ is required is equal to t_0 resp. $t_0 + \tau_{\text{osm}}$, i.e. that

$$t_1 - \tau_0 + c^-\tau_0 = t_0 \quad \text{and} \quad t_1 - \tau_0 + c^+\tau_0 = t_0 + \tau_{\text{osm}}$$

holds. This is a linear system for t_1 and τ_0 with unique solution

$$t_1 = t_0 + \frac{1 - c^-}{c^+ - c^-}\tau_{\text{osm}} \quad \text{and} \quad \tau_0 = \frac{1}{c^+ - c^-}\tau_{\text{osm}} .$$

Now we define, using the output of the continuous one-step method,

$$Y_{0i} := \tilde{y}(t_1 - \tau_0 + c_i\tau_0) = \tilde{y}\left(t_0 + \frac{c_i - c^-}{c^+ - c^-}\tau_{\text{osm}}\right) , \quad i = 1, 2, \ldots, s .$$

Note that in particular the prescribed initial data y_0 of the ODE is included as one vector of Y_0. For simplicity we finally choose $\tau_1 = \tau_0$, i.e. $\sigma_1 = 1$.

3 Implementation of two-step peer methods in KARDOS

3.1 Problem class

KARDOS is used to solve time-dependent systems of n PDEs or PDAEs in a spatial domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$. We consider here the case $\Omega \subset \mathbb{R}^2$. The solution is denoted by $u = (u_1, \ldots, u_n)$. The equation i of the system must be of the form

$$\sum_{j=1}^{n} P_{ij}\partial_t u_j - \nabla \cdot \left[\sum_{j=1}^{n} \begin{pmatrix} L_{ij}^{xx} & L_{ij}^{xy} \\ L_{ij}^{yx} & L_{ij}^{yy} \end{pmatrix} \nabla u_j\right] = S_i .$$

Here, P is the "parabolic" matrix, $L^{xx}, L^{xy}, L^{yx}, L^{yy}$ are the diffusion coefficient matrices, and S is the source term vector of the problem which can also contain first order derivatives of the solution modelling

advection. The matrices are all $n \times n$ matrices, the source term vector has length n, and all these user-supplied coefficients can be functions of time, space, state u, and the spatial derivatives of the state, ∇u. Matrix P can be singular which renders the problem a PDAE. For this work we assume that matrix P does not depend on time, state or derivatives of the state. For each component of the system a consistent initial condition must be supplied and a boundary condition on each boundary point of the domain be prescribed. KARDOS supports three kinds of boundary conditions. Dirichlet boundary conditions are implemented as

$$g_i(t, x, y, u(t, x, y)) = 0$$

with a user specified function g_i.

Cauchy boundary conditions for the flux of the i-th equation are specified as

$$n(x, y) \cdot \left[\sum_{j=1}^{n} \begin{pmatrix} L_{ij}^{xx} & L_{ij}^{xy} \\ L_{ij}^{yx} & L_{ij}^{yy} \end{pmatrix} \nabla u_j \right] = g_i(t, x, y, u(t, x, y))$$

with a user specified function g_i and $n(x, y)$ the unit outward normal vector. Neumann (no flux) boundary conditions are the special case $g_i = 0$ of Cauchy boundary conditions.

3.2 Formulation as an abstract ODE problem and finite element formulation

We write the PDE system which is to be solved with KARDOS in the form of an abstract ODE/DAE in a Hilbert space setting, see [6],

$$P\dot{u} = f(u(t)), \quad u(t_0) = u_0, \tag{9}$$

where the right-hand side f, for simplicity again only autonomous, now is a differential operator containing the diffusion and source terms which are present in the general problem class specification, see Sec. 3.1, and P is the "parabolic" matrix of the problem. We remind the reader that in this work we assume that P does not depend on time, state or derivatives of the state.

We can now formally apply the linearly implicit two-step peer method (6a,6b,8) to the abstract DAE problem (9). The space dependent stage value $u_{mi}(\cdot) \approx u(t_{mi}, \cdot)$ is then obtained from the linear (PDE) system, see (8),

$$\left(\frac{P}{\tau_m \gamma} - T_m \right) \underbrace{(u_{mi} - u_{mi}^0)}_{=:\delta_{mi}} = f(u_{mi}^0) + P \underbrace{\frac{1}{\tau_m \gamma} (w_i - u_{mi}^0)}_{=:g_{mi}} \tag{10}$$

with a definition of w_i and u_{mi}^0 according to (6a) and (6b). As before, matrix T_m is constant for each time step and is an approximation to the Jacobian matrix of f w.r.t. u evaluated at $u_{m-1,s}$.

We apply a Galerkin finite element scheme for the solution of the time-independent problem (10). Let $\{\phi_j(x, y) : j \in \mathcal{J}\}$ be the system of FE basis functions on the (fixed) FE grid $\mathcal{G}$, where $\mathcal{J}$ is the corresponding index set. The function $u_{mi}(x, y)$ is now replaced by the FE representation

$$u_{h,mi}(x, y) = \sum_{j \in \mathcal{J}} u_{h,mi}^j \phi_j(x, y)$$

with unknown real coefficients $u_{h,mi}^j$ which are collected to the vector $U_{h,mi}$. Accordingly, for other functions like $\delta_{mi}(x, y)$, $g_{mi}(x, y), \ldots$ the corresponding FE representation $\delta_{h,mi}(x, y)$, $g_{h,mi}(x, y), \ldots$ and vectors $\Delta_{h,mi}$, $G_{h,mi}, \ldots$ are introduced. Multiplication of (10) by a test function $\phi_l(x, y)$, $l \in \mathcal{J}$, and integration over the spatial domain yields

$$\left\langle \left(\frac{P}{\tau_m \gamma} - T_m \right) \sum_{j \in \mathcal{J}} \delta_{h,mi}^j \phi_j, \phi_l \right\rangle = \left\langle f(u_{h,mi}^0) + P \sum_{j \in \mathcal{J}} g_{h,mi}^j \phi_j, \phi_l \right\rangle . \tag{11}$$

Employing the linearity of the integral, this can be rewritten as

$$\sum_{j\in\mathcal{J}} \delta^j_{h,mi} \left(\frac{1}{\tau_m\gamma} \underbrace{\langle P\phi_j, \phi_l\rangle}_{=:\mathcal{P}_{lj}} - \underbrace{\langle T_m\phi_j, \phi_l\rangle}_{=:\mathcal{S}_{lj}} \right) = \underbrace{\left\langle f(u^0_{h,mi}), \phi_l \right\rangle}_{=:\mathcal{F}_l} + \sum_{j\in\mathcal{J}} g^j_{h,mi} \langle P\phi_j, \phi_l\rangle \,. \tag{12}$$

Here $\mathcal{P} = (\mathcal{P}_{lj})$ is called the mass matrix in the case $P = I$, $\mathcal{S} = (\mathcal{S}_{lj})$ is the stiffness matrix, and $\mathcal{F} = (\mathcal{F}_l)$ is the load vector of the FE approximation. Summarising the equations for all basis functions ϕ_l, $l \in \mathcal{J}$, yields finally the linear equation system

$$\left(\frac{1}{\tau_m\gamma}\mathcal{P} - \mathcal{S} \right) \Delta_{h,mi} = \mathcal{F} + \mathcal{P} G_{h,mi} \,. \tag{13}$$

This linear system is assembled and solved for $\Delta_{h,mi}$ in the KARDOS system employing either direct or iterative (with possible preconditioning) solvers. Finally, the coefficients of the predictor are updated to yield the coefficients of the stage solution on the FE grid $\mathcal{G}$, that is $U_{h,mi} = U^0_{h,mi} + \Delta_{h,mi}$. This is equivalent to $u_{h,mi}(x,y) = u^0_{h,mi}(x,y) + \delta_{h,mi}(x,y)$ in Ω.

3.3 Time-step control

Let $u_h(t_m,\cdot,\cdot) = (u_{h,1}(t_m,\cdot,\cdot),\ldots,u_{h,n}(t_m,\cdot,\cdot))$ denote the computed numerical solution at $t = t_m$ and $\tilde{u}_h(t_m,\cdot,\cdot) = (\tilde{u}_{h,1}(t_m,\cdot,\cdot),\ldots,\tilde{u}_{h,n}(t_m,\cdot,\cdot))$ the corresponding embedded solution. Then the estimated local temporal error of the time step is

$$le_m(\cdot,\cdot) := u_h(t_m,\cdot,\cdot) - \tilde{u}_h(t_m,\cdot,\cdot) \,.$$

We measure this local error in a weighted root mean square norm, that is we define

$$ERR_t := \left(\frac{1}{n} \sum_{i=1}^{n} \frac{\|le_m(\cdot,\cdot)\|^2_{L_2}}{(ScalR_i \cdot \|u_{h,i}(t_m,\cdot,\cdot)\|_{L_2} + ScalA_i \cdot \sqrt{|\Omega|})^2} \right)^{\frac{1}{2}} ,$$

where $ScalR_i$ and $ScalA_i$ are user-prescribed relative and absolute scaling factors for each component of the PDE system. The user further prescribes a temporal tolerance TOL_t. Since we currently ignore spatial discretisation errors, a time step is accepted if $ERR_t < TOL_t$ and otherwise rejected. In any case, the step size prediction for the following time step is computed as

$$\tau_{\text{new}} = \min\{\tau_{\max}, \min\{\alpha_{\max}, \max\{\alpha_{\min}, (TOL_t/ERR_t)^{1/(\tilde{p}+1)}\}\} \cdot \alpha_{\text{safe}} \cdot \tau_m\}$$

following a standard step size control with parameters $\alpha_{\min}, \alpha_{\max}, \alpha_{\text{safe}}$, and $\tau_{\max}$. Furthermore, $\tilde{p}$ is the order of the embedded method. In the case of the two-step peer methods described in this work we have $\tilde{p} = s - 2$ and we further use rather strict values

$$\alpha_{\min} = 0.2 \,, \quad \alpha_{\max} = 2 \,, \quad \alpha_{\text{safe}} = 0.9$$

to ensure a stable change of time step sizes. For one-step methods we use the more relaxed standard values of KARDOS.

4 Numerical experiments

All experiments are performed on fixed spatial finite element meshes with number of nodes as detailed below. First, second, and third order Lagrange finite elements are used depending on the problem at hand. All linear systems (13) are solved with BiCGStab with ILU preconditioner. In the following numerical experiments the tolerance level for the linear system solver is set to very strict values. This is done on purpose because our main interest is in the numerical investigation of the error in the time integration schemes and we would like to avoid an influence of the linear system solves on these errors.

We use the two-step methods `peer4`, `peer5`, and `peer6` as defined in Section 2.1. We compare these newly implemented methods in KARDOS with the established one-step time integration methods `ros3p` [9], `rodas` [4], and `rodasp` [18, 19]. The 3-stage method `ros3p` has classical and stiff order three. The 6-stage methods `rodas` and `rodasp` have classical order four. Whereas `rodasp` has also order four for linear stiff problems, method `rodas` generally suffers from order reduction when applied to stiff problems.

As before, let $u_h = (u_{h,1}, \dots, u_{h,n})$ denote the computed numerical solution. At each integration time point $t_m \in (t_0, t_{\text{end}}]$ we can compare the numerical solution $u_h(t_m, x, y)$ with a reference solution $\hat{u} = (\hat{u}_1, \dots, \hat{u}_n)$. The reference solution can either be the exact solution $u = (u_1, \dots, u_n)$ at t_m or, for instance, a solution computed with higher accuracy. We use the L_2-norm to compare u_h with $\hat{u}$ at t_m, i.e.

$$\|u_h(t_m, \cdot, \cdot) - \hat{u}(t_m, \cdot, \cdot)\|_{L_2}^2 := \frac{1}{n} \sum_{i=1}^{n} \|u_{h,i}(t_m, \cdot, \cdot) - \hat{u}_i(t_m, \cdot, \cdot)\|_{L_2}^2 \,.$$

The integral involved in the norm is approximated using a high-order quadrature formula. If an analytic solution u of a problem is available we also report the $L_2 - L_2$-error in space *and* time of a computed numerical solution u_h with respect to u. Here, the integral in time is approximated with first order leading to the definition

$$\|u_h - u\|_{L_2-L_2}^2 := \sum_{m=1}^{M} \tau_{m-1} \|u_h(t_m, \cdot, \cdot) - u(t_m, \cdot, \cdot)\|_{L_2}^2 \,,$$

where M is the number of time steps taken (including the starting step for the two-step methods). All errors considered here are absolute errors.

4.1 Burgers equation

As a first example we consider on the unit square $\Omega = (0, 1)^2$ and for times $t \in (0, t_{\text{end}}]$ the system of two Burgers equations [6, p. 53]

$$u_t = \nabla \cdot (D \nabla u) - a(u u_x + v u_y) \tag{14a}$$

$$v_t = \nabla \cdot (D \nabla v) - a(u v_x + v v_y) \tag{14b}$$

with parameter values

$$D = 0.01 \,, \quad a = 1 \,, \quad t_{\text{end}} = 2 \tag{14c}$$

and exact solution

$$u(t, x, y) = \frac{3}{4} - \frac{1}{4a} \left(1 + \exp \frac{-4x + 4y - t}{32D} \right)^{-1} , \tag{14d}$$

$$v(t, x, y) = \frac{3}{4} + \frac{1}{4a} \left(1 + \exp \frac{-4x + 4y - t}{32D} \right)^{-1} . \tag{14e}$$

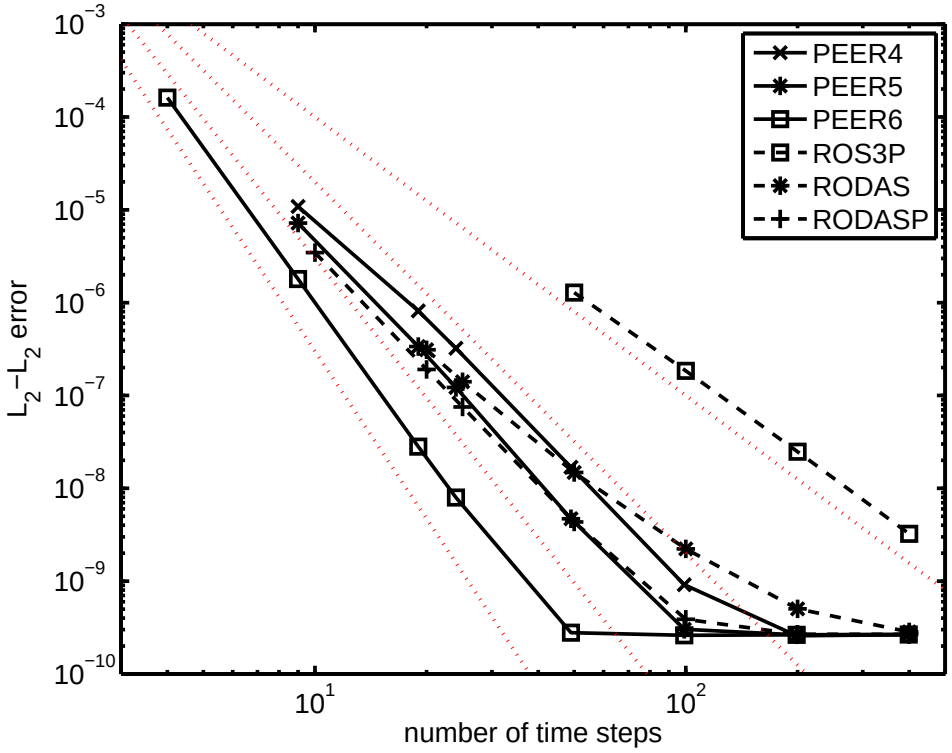

Figure 1: Number of time steps vs. $L_2 - L_2$-error plots for the constant time step computations up to $t_{end} = 2$ for the Burgers equation. Computations have been performed with 5, 10, 20, 25, 50, 100, 200, and 400 time steps. Missing markers indicate that a method did not complete the computation successfully with that number of steps. The dotted lines are the order slopes for orders $3, 4, 5$, and 6.

The initial and Dirichlet boundary conditions are taken from the exact solution. Each exact solution depends in space only on the difference $x - y$ and hence represents a wave starting on the $x = y$ diagonal of the domain and moving towards its north-west corner.

The following computations for this problem are performed with cubic finite elements on a constant spatial grid with 8321 nodes. This leads to linear equation systems of dimension 148226 which are solved iteratively up to a scaled residual tolerance of 10^{-10}.

4.1.1 Computations with fixed time step size

We perform fixed time step size computation for this problem up to time $t_{\text{end}} = 2$ to demonstrate the orders of convergence in time of the various methods. Since we have the exact solution of this problem available, we start the computations of the two-step methods on that exact solution, i.e. Y_{0i}, $i = 1 \ldots, s$ is taken from the exact solution. The $L_2 - L_2$-errors in the numerical solution w.r.t. the analytic solution of the problem are presented in Fig. 1 and discussed below.

The first observation from Fig. 1 is that for this problem the error due to the approximation in space with cubic finite elements on the selected grid is about $5 \cdot 10^{-10}$.

The one-step methods `ros3p` and `rodasp` show orders of convergence three resp. four. The observed order for method `rodas` is close to three and so less than the order of the method. This is a case of order reduction.

Concerning the two-step methods, they should exhibit super-convergence properties in fixed step size computations with order of convergence equal to the number of stages. For methods `peer4` and

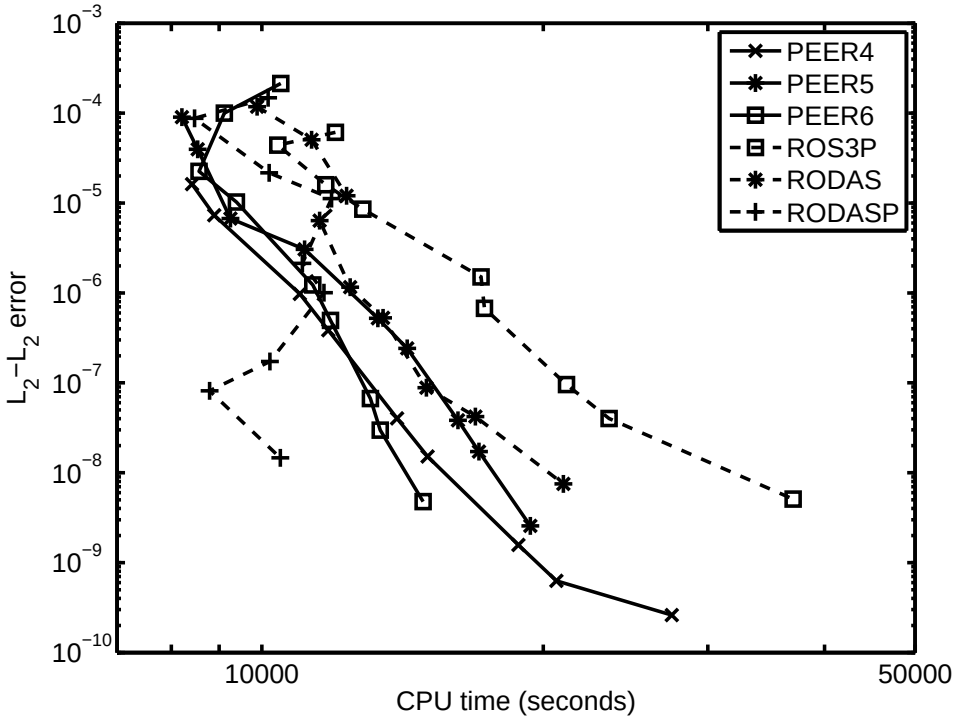

Figure 2: CPU time vs. $L_2 - L_2$-error plots for the variable time step computations up to $t_{end} = 2$ for the Burgers equation. The requested temporal accuracies are $TOL_t = 10^{-3}, 5 \cdot 10^{-4}, 10^{-4}, \ldots, 10^{-7}$ and the initial step has a length of $\tau_{\text{osm}} = \max\{5 \cdot 10^{-4}, 100 \cdot TOL_t\}$. All methods are able to compute the numerical solution for all values of TOL_t.

`peer6` this is clearly the case before they reach the limiting error level of $5 \cdot 10^{-10}$. For method `peer5` the situation is not so obvious and the observed order is somewhere between four and five.

4.1.2 Computations with variable time step size

We perform variable time step size computation for this problem up to time $t_{\text{end}} = 2$ to investigate the computational efficiency of the various methods. The two-step methods are started with one time-step of the one-step method `ros3p`. The $L_2 - L_2$-errors in the numerical solution w.r.t. the analytic solution of the problem are presented in Fig. 2 and discussed below.

The variable time step size computations presented in Fig. 2 show that the two-step methods perform better than the one-step methods `ros3p` and `rodas`; method `rodasp` performs slightly better than the two-step methods for this example but at the price of a hardly predictable dependence between achieved error and required CPU time. The higher order two-step methods exhibit some advantage over the four-stage method `peer4` for stricter tolerance requirements but there is no real gain for this particular example.

4.2 PDAE problem

The following partial differential-algebraic equation system is taken from [15, pp. 89]. We consider in the spatial domain $\Omega = (0,1)^2$ and for times $t \in (0, t_{\text{end}}]$ the system

$$u_t - \Delta u - \Delta v + x u_x + y u_y - u + v = f_1\,, \tag{15a}$$

$$-\Delta u - \Delta v + u^3 + v^3 = f_2\,. \tag{15b}$$

The initial and boundary conditions as well as the functions f_1 and f_2 are taken from the exact solution

$$u(t,x,y) = (2x+y)\sin(t)\,, \quad v(t,x,y) = (x+3y)\cos(t)\,. \tag{15c}$$

This system has perturbation index 1, see [15].

The exact solution is linear in space and so can be represented exactly by using linear finite elements. Hence we consider all errors in the numerical solution to come from the time stepping scheme. The following computations for this problem are performed with linear finite elements on constant spatial grid with 1089 nodes. This leads to linear equation systems of dimension 2178. These linear systems are solved iteratively up to a scaled residual tolerance of 10^{-14} in the constant time step case and 10^{-12} in the variable time step case.

4.2.1 Fixed time step sizes

We perform fixed time step size computation for this problem up to time $t_{\text{end}} = 1$ to demonstrate the orders of convergence in time of the various methods. Since we have the exact solution of this problem available, we start the computations of the two-step methods on that exact solution. The $L_2 - L_2$-errors in the numerical solution w.r.t. the analytic solution of the problem are presented in Fig. 3 and discussed below.

The plots in Fig. 3 demonstrate that for this example the one-step methods show their expected order of convergence. It is noteworthy that they cannot achieve an accuracy higher than $\approx 10^{-10}$.

The two-step methods, in particular `peer4` and `peer6` and to a lesser extend also `peer5`, exhibit the expected super convergence in fixed step size computations, that is their observed order of convergence equals their number of stages. The two-step methods can compute approximations to the exact solution up to an error of 10^{-13}, i.e. almost down to the rounding error. Why the errors increase again for smaller time steps is unclear. Note that this behaviour shows up for one-step methods also in the results presented by Rang [15]. A possible cause for this problem could be a potential instability to perturbations of the solution of this problem.

4.2.2 Variable time step sizes

We perform variable time step size computation for this problem up to time $t_{\text{end}} = 1$ to investigate the computational efficiency of the various methods. The two-step methods are started with one time-step of the one-step method `ros3p`. The $L_2 - L_2$-errors in the numerical solution w.r.t. the analytic solution of the problem are presented in Fig. 4. The improved efficiency of the two-step methods compared to the one-step methods is clearly visible from these plots. Furthermore, for higher accuracy demands, the application of methods of higher order is beneficial.

4.3 Gray-Scott model

The Gray-Scott model [12] is a 2D pattern formation model of reaction–diffusion type. We consider the model on $\Omega = (0, 2.5)^2$ and for $t \in [0, t_{\text{end}}]$. The chemical concentrations u and v satisfy the PDE

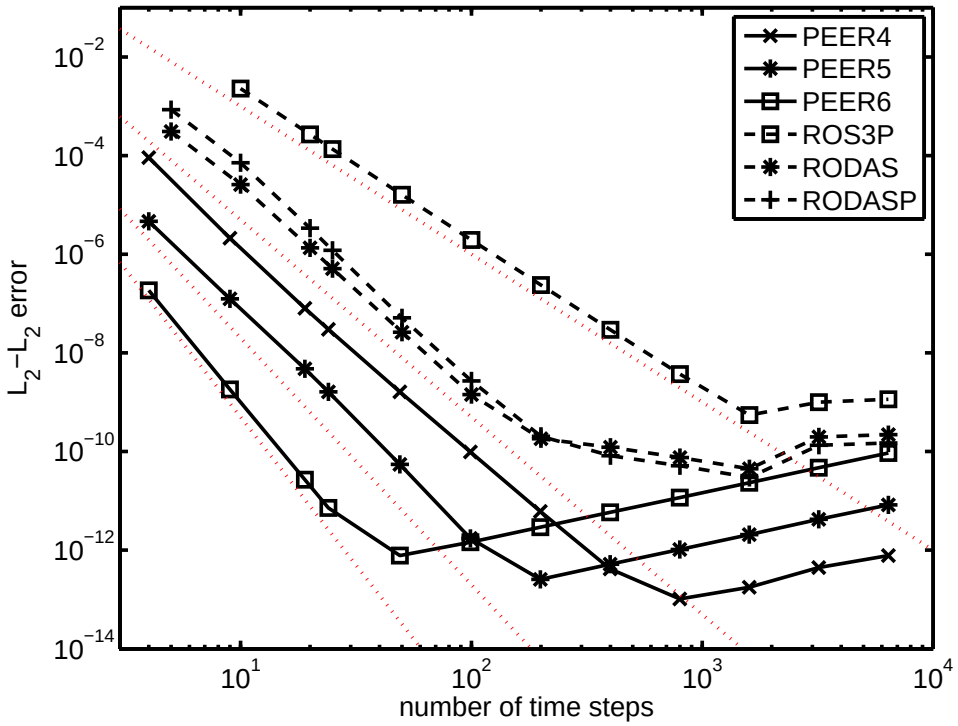

Figure 3: Number of time steps vs. $L_2 - L_2$-error plots for the constant time step computations up to $t_{end} = 1$ for problem PDAE. Computations have been performed with 5, 10, 20, 25, 50, 100 to 6400 time steps. All computations completed successfully except for `ros3p` with 5 time steps. The dotted lines are the order slopes for orders $3, 4, 5$, and 6.

system and initial conditions given by

$$u_t = \nabla \cdot (D_1 \nabla u) - uv^2 + \gamma(1-u)\,, \tag{16a}$$

$$v_t = \nabla \cdot (D_2 \nabla v) + uv^2 - (\gamma+\kappa)v\,, \tag{16b}$$

$$u(x,y,0) = 1 - 2v(x,y,0)\,, \tag{16c}$$

$$v(x,y,0) = \begin{cases} \frac{1}{4}\sin^2(4\pi x)\sin^2(4\pi y) & : \quad 1 \le x,y \le 1.5 \\ 0 & : \quad \text{otherwise} \end{cases}, \tag{16d}$$

together with zero flux boundary conditions. The parameter values are chosen as in [5, pp. 21] and given by

$$t_{\text{end}} = 1000\,, \quad D_1 = 8 \cdot 10^{-5}\,, \quad D_2 = 4 \cdot 10^{-5}\,, \quad \gamma = 0.024\,, \quad \kappa = 0.06\,.$$

No analytic solution is available for this problem. This problem is non-stiff and hence can be solved efficiently with, e.g. a finite difference approximation on a fixed spatial grid and explicit time stepping methods. However, we include this example here (and use linearly implicit time stepping schemes) because in order to achieve a high accuracy we will in future work use spatially adaptive FE meshes. The FEM leads to coupled linear systems (13) even in the case that we chose the Jacobian approximation T to be the zero matrix except if we use mass lumping. So there is no advantage to use explicit schemes over linearly implicit schemes.

We use KARDOS with quadratic finite elements on a regular mesh with 4225 nodes to solve this problem. This leads to linear equation systems of dimension 33282. We evaluate the numerical solutions

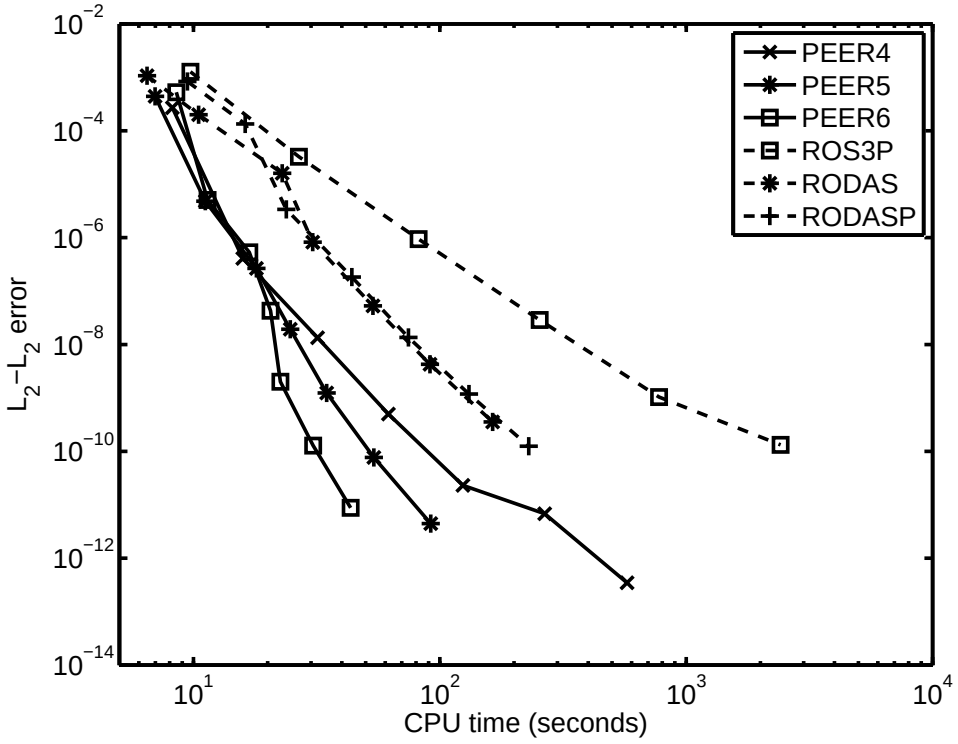

Figure 4: CPU time vs. $L_2 - L_2$-error plots for the variable time step computations up to $t_{end} = 1$ for problem PDAE. The requested temporal accuracies are $TOL_t = 10^{-k}$ for $k = 3, 4, \ldots, 9$ and the initial step has a length of $\tau_{\text{osm}} = \max\{5 \cdot 10^{-4}, 100 \cdot TOL_t\}$. All methods are able to compute the numerical solution for all values of TOL_t.

computed at final time $t_{\text{end}} = 1000$ with respect to a reference solution obtained from a high temporal accuracy computation (on the same spatial grid) with time stepping using method `rodasp`. The L_2-error at final time vs. CPU time plots are shown in Fig. 5.

The plots in Fig. 5 show that the two-step methods tested are superior to the one-step methods of order three and four in KARDOS for the problem under investigation. Furthermore, we see that solutions of higher accuracy are computed in less time with increasing order of the peer methods. Higher-order peer methods certainly pay off for this example.

5 Conclusions and outlook

We have introduced linearly-implicit two-step peer methods as time integration schemes in the FE software KARDOS. In the numerical tests on fixed spatial grids we have observed no order reduction with these methods. The two-step peer methods are very accurate in comparison with the tested one-step methods. However, they are sensitive with respect to inaccurate starting values. This often makes a rather small initial time step size necessary in variable time step computations. Despite this drawback they are more efficient, at least competitive, in comparison with the tested one-step methods for the examples considered in this work.

We are currently investigating the use of linearly-implicit two-step peer methods in combination with spatial grid adaption in KARDOS. This requires the investigation of particular difficulties of the two-step methods, for instance the control of the local temporal error and the choice of suitable time

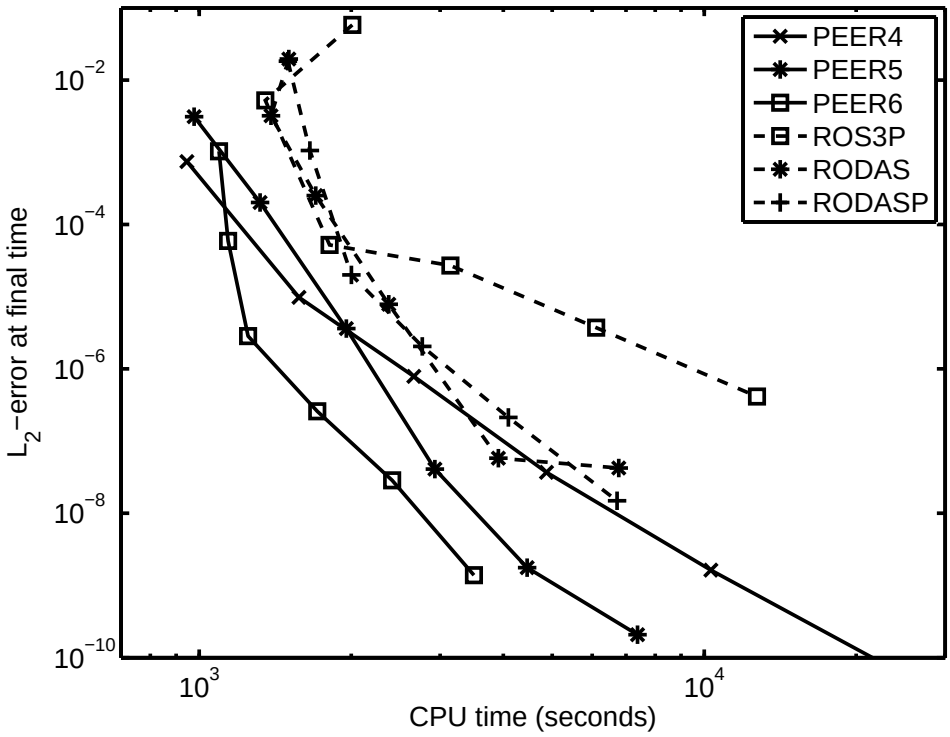

Figure 5: CPU time vs. $L_2 - L_2$-error plots for the variable time step computations up to $t_{end} = 1000$ for Gray-Scott problem. The requested temporal accuracies are $TOL_t = 10^{-k}$ for $k = 3, 4, \ldots, 8$. The initial step has a length of $h_{\text{osm}} = TOL_t$. All methods are able to compute the numerical solution for all values of TOL_t.

step sizes, when the spatial grid changes from time step to time step.

References

[1] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2003.

[2] B. Erdmann, J. Lang, and R. Roitzsch. KARDOS user's guide. Technical Report ZR 02-42, Konrad-Zuse-Zentrum Berlin, 2002.

[3] J. Fröhlich and J. Lang. Two-dimensional cascadic finite element computations of combustion problems. *Comp. Meth. Appl. Mech. Eng.*, 158:255–267, 1998.

[4] E. Hairer and G. Wanner. *Solving ordinary differential equations. II. Stiff and differential-algebraic problems*. Number 14 in Springer Series in Computational Mathematics. Springer–Verlag, Berlin, 2nd edition, 1996.

[5] W. Hundsdorfer and J. G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, volume 33 of *Springer Series in Computational Mathematics*. Springer–Verlag Berlin, Heidelberg, New York, 2003.

[6] J. Lang. *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems*. Number 16 in Lecture Notes in Computational Science and Engineering. Springer-Verlag Berlin Heidelberg, 2001.

[7] J. Lang, B. Erdmann, and M. Seebass. Impact of nonlinear heat transfer on temperature control in regional hyperthermia. *IEEE Trans. Biomedical Engrg.*, 46:1129–1138, 1999.

[8] J. Lang and W. Merz. Two-dimensional adaptive simulation of dopant diffusion in silicon. *Computing and Visualization in Science*, 3:169–176, 2001.

[9] J. Lang and J. Verwer. ROS3P—an accurate third-order Rosenbrock solver designed for parabolic problems. *BIT*, 41:731–738, 2001.

[10] Ch. Lubich and A. Ostermann. Linearly implicit time discretization of non-linear parabolic equations. *IMA J. Numer. Anal.*, 15:555–583, 1995.

[11] Ch. Lubich and A. Ostermann. Runge-Kutta approximation of quasi-linear parabolic equations. *Math. Comput.*, 64:601–627, 1995.

[12] J. E. Pearson. Complex patterns in a simple system. *Science*, 261:189–192, 1993.

[13] H. Podhaisky, R. Weiner, and B. A. Schmitt. Numerical experiments with parallel two-step W-methods for DAEs. *PAMM*, 1:512–513, 2002.

[14] H. Podhaisky, R. Weiner, and B. A. Schmitt. Rosenbrock-type 'Peer' two-step methods. *Appl. Numer. Math.*, 53:409–420, 2005.

[15] J. Rang. *Stability estimates and numerical methods for degenerate parabolic differential equations*. PhD thesis, Technische Universität Clausthal, 2004.

[16] B. A. Schmitt and R. Weiner. Parallel two-step W-methods with peer variables. *SIAM Journal on Numerical Analysis*, 42:265–282, 2004.

[17] B. A. Schmitt, R. Weiner, and H. Podhaisky. Multi-implicit peer two-step W-methods for parallel time integration. *BIT Numerical Mathematics*, 45:197–217, 2005.

[18] G. Steinebach. Order-reduction of ROW-methods for DAEs and method of lines applications. Technical Report 1741, Technische Hochschule Darmstadt, Germany, 1995.

[19] G. Steinebach and P. Rentrop. An adaptive method of lines approach for modelling flow and transport in rivers. In A. Vande Wouver, Ph. Sauces, and W. E. Schiesser, editors, *Adaptive method of lines*, pages 181–205. Chapman & Hall/CRC, 2001.

[20] R. Weiner, B. A. Schmitt, and H. Podhaisky. Parallel 'Peer' two-step W-methods and their application to MOL-systems. *Appl. Numer. Math.*, 48:425–439, 2004.