Numerische Mathematik manuscript No.
(will be inserted by the editor)

Efficient numerical realization of discontinuous Galerkin methods for temporal discretization of parabolic problems

Thomas Richter · **Andreas Springer** ·
Boris Vexler

Received: date / Accepted: date

Abstract We present an efficient and easy to implement approach to solving the semidiscrete equation systems resulting from time discretization of nonlinear parabolic problems with discontinuous Galerkin methods of order r. It is based on applying Newton's method and decoupling the Newton update equation, which consists of a coupled system of $r+1$ elliptic problems. In order to avoid complex coefficients which arise inevitably in the equations obtained by a direct decoupling, we decouple not the exact Newton update equation but a suitable approximation. The resulting solution scheme is shown to possess fast linear convergence and consists of several steps with same structure as implicit Euler steps. We construct concrete realizations for order one to three and give numerical evidence that the required computing time is reduced significantly compared to assembling and solving the complete coupled system by Newton's method.

Keywords discontinuous Galerkin method · parabolic equations · inexact Newton's method

Mathematics Subject Classification (2000) 65M99 · 65M60 · 65F10

T. Richter
Institut für angewandte Mathematik, Ruprecht-Karls-Universität Heidelberg, Im Neuenheimer Feld 294, 69120 Heidelberg, Germany
E-mail: richter@uni-hd.de

A. Springer
Lehrstuhl für Mathematische Optimierung, Technische Universität München, Fakultät für Mathematik, Boltzmannstraße 3, 85748 Garching b. München, Germany
Tel.: +49-89-289-17993
Fax: +49-89-289-17932
E-mail: springer@ma.tum.de

B. Vexler
Lehrstuhl für Mathematische Optimierung, Technische Universität München, Fakultät für Mathematik, Boltzmannstraße 3, 85748 Garching b. München, Germany
E-mail: vexler@ma.tum.de

1 Introduction

In this paper we propose an efficient numerical realization of discontinuous Galerkin methods (dG) for temporal discretization of parabolic partial differential equations. In contrast to standard time-stepping schemes (like Runge-Kutta methods) the dG schemes are based on the space-time weak formulation of parabolic equations and employ a Galerkin finite element procedure to obtain a (semi-)discrete system of equations. Although these methods are formulated globally in the space-time domain, the resulting (semi-)discrete system decouples into a sequence of systems corresponding to individual time steps. This equivalence to a time stepping scheme is due to the discontinuity of the used test space. Discontinuous Galerkin type methods were first investigated for solving the Neutron transport equation (see for example [14]), the time stepping schemes of this type for parabolic equations were developed by Johnson, Thomee and coworkers, see, e. g., [9,7,8,27].

There is a number of important properties making the use of dG schemes attractive for temporal discretization of parabolic equations:

- *A priori error analysis.* This type of discretization allows for a priori error estimates of optimal order with respect to discretization parameters (as the size of time steps) as well as with respect to the regularity requirements for the solution, see, e. g., [7,8].
- *A posteriori error analysis and adaptivity*. Different systematic approaches for a posteriori error estimation and adaptivity developed for finite element discretizations can be adapted for dG temporal discretization of parabolic equations, see, e. g. [24,22]
- *Hp time stepping.* Compared to the traditional approach to adaptivity that refines the length of the considered time intervals (h-adaptivity), the hp approach allows also to increase the order on selected intervals which results in faster convergence. For dG methods, this approach is developed for example in [24].
- *Dynamic meshes.* Since the trial space allows for discontinuities at the time nodes, the use of different spatial discretizations for each time step can be directly incorporated into the discrete formulation, see, e. g. [22].
- *Discretization of optimal control problems.* For the treatment of optimal control problems, Galerkin methods are particularly suitable since they expose the important property that the two approaches *optimize-then-discretize*, i.e. the discretization of the optimality system built up on the continuous level, and *discretize-then-optimize*, i.e. discretization of the state equation and subsequent construction of the optimality system on the discrete level, lead to the same discretization scheme, see, e. g., [1]. Compared to continuous Petrov-Galerkin time-stepping schemes, see [18], dG schemes also have the advantage that the adjoint state can use the same discretization as the state variable. This allows for unified numerical treatment and simplifies a priori and a posteriori error analysis, see, e. g. [15–17,4].
- *Stability properties.* Compared to, e. g., the continuous Galerkin methods, dG schemes are not only A-stable but strongly A-stable, see, e. g., [14].

– *Nodal superconvergence.* If the solution to the problem has sufficient regularity, the discontinuous Galerkin method of order r will expose superconvergence of order $2r+1$ at the nodal points, see, e. g., [27, Chapter 12].

Despite those advantages, dG schemes of higher order than zero are seldom used for practical computations. Two important reasons for this are the comparatively high effort required to implement them and the computational cost for solving the time stepping equation. Since the number of unknowns per time step grows linearly with increasing order, assembling and solving the resulting coupled equation system can become prohibitive, especially in the context of large systems of nonlinear PDEs in several space dimensions. Results for first order dG schemes where the full time stepping equations were solved can be found for example in [12] for the heat equation and for Navier-Stokes in [2].

For linear equations with time-independent coefficients, an efficient solution approach was developed by Schötzau, Schwab and their coworkers in the papers [23] and [30]. There, the dG equation system was decoupled into linear problems with the same structure as the time-stepping equation for the implicit Euler scheme. However, the resulting equations are complex-valued which on the one hand means considerable computational effort, on the other hand can make implementation more difficult since not every existing finite element code will support complex numbers.

Here, we analyze in more detail why conventional decoupling attempts always lead to complex-valued equations and present a novel approach based on an inexact Newton iteration which circumvents this problem. We perform a block elimination on the dG system to transform it into upper triangular form and apply an approximation such that the resulting equation system can be solved efficiently without requiring complex arithmetic. To compensate for the error caused by the approximation, we iterate our scheme in a Newton-like fashion. For linear problems with time-independent coefficients the overhead caused by the iterations will usually outweigh the time savings from avoiding solving the full dG equation. In the nonlinear case however, Newton steps are required for the exact scheme as well and the necessary reassembling of the system matrix occupies a significant part of the overall computing time. So particularly for those difficult nonlinear problems the simplified schemes proposed here will reduce the overall run time significantly. Additionally memory consumption is reduced and the implementation is simplified since the matrix for the full dG time stepping equation does not need to be assembled. As for the approach in [23], all sub-steps consist of equations with the same structure as the implicit Euler scheme. Hence, our method can be realized on top of any existing finite element code that allows for an implementation of the backward Euler scheme.

The outline of the paper is as follows: In Section 2 we give a detailed definition for the considered class of parabolic equations and show two sets of assumptions on the spatial differential operator that guarantee the existence of a unique solution and the applicability of the proposed simplified solution scheme. We discretize the problem in Section 3 with respect to time and space. Section 4 focuses on solving the discretized problem by Newton's method and on the structure of the resulting linear sub-problems. In Section 5 we analyze why this structure makes it impossible to decouple the linear sub-problem without introducing complex-valued equations.

Subsequently we show how to derive a class of approximate solution schemes for the Newton update equation which circumvent this problem. The accuracy of these approximations and their impact on the outer Newton iteration are investigated in Section 6. Section 7 gives realizations of such schemes for order 1 to 3. Finally, in Section 8 we study the efficiency of the presented schemes by numerical tests.

2 Problem statement

For a given time interval $I = (0,T)$, we consider parabolic equations of the form

$$\begin{aligned} \partial_t u(t) + A(u) &= f(t) \quad \forall t \in I, \\ u(0) &= u_0. \end{aligned}$$

To state the precise functional analytic setting for our problem, let the Hilbert space V be continuously and densely embedded into the Hilbert space $H := L^2(\Omega)$ where $\Omega \subseteq \mathbb{R}^d$ with $d = 2$ or $d = 3$ is a polygonal spatial domain. After identifying H with its dual space H^*, the spaces $V \hookrightarrow H \hookrightarrow V^*$ form a Gelfand triple. Let $A : V \to V^*$ be a possibly nonlinear elliptic partial differential operator. We require A to be Fréchet differentiable and the derivatives $A_u(u) : V \to V^*$ to be self-adjoint and positive when considered as unbounded operators on H for all $u \in V$. At the end of this section we give examples for operators satisfying those conditions.

On the time interval I we introduce the abstract function space

$$X := W(I) = \left\{ v \in L^2(I,V) \middle| \partial_t v \in L^2(I,V^*) \right\}.$$

We denote the inner product on H by $(\cdot,\cdot)$ and the inner product on $L^2(I,H)$ by $(\cdot,\cdot)_I$. With these preparations we can state a weak formulation of the considered class of equations: Find $u \in X$ satisfying

$$\int_I \langle \partial_t u, \Phi \rangle_{V^* \times V} \,\mathrm{d}t + \int_I a(u)(\Phi) \,\mathrm{d}t + (u(0), \Phi(0)) = (f, \Phi)_I + (u_0, \Phi), \quad \forall \Phi \in X, \tag{1}$$

where $\langle \cdot,\cdot \rangle_{V^* \times V}$ denotes the duality pairing on V. The semilinear form a (linear in the second argument) is defined by

$$a(u)(\cdot) := \langle A(u), \cdot \rangle_{V^* \times V}.$$

For the initial value we require $u_0 \in H$ and for the right hand side $f \in L^2(I,V^*)$. Under which conditions this equation attains a solution depends on the form of the operator A. Throughout this paper we will restrict our analysis to problems that possess a unique solution.

There are several different sets of conditions that will ensure on the one hand that the operator A fulfills the stated requirements and on the other hand that the continuous problem and its discretization are well-posed. Examples for such sets of conditions include

(I) A is a linear operator of the form $A(u) := -\operatorname{div}(G\nabla u) + \gamma u$ where G is a symmetric $r \times r$ matrix with entries $G_{ij} \in L^\infty(\Omega)$ which is uniformly positive definite on Ω. We further assume $\gamma \in L^\infty(\Omega)$. The space V can be chosen either as $V := H_0^1(\Omega)$ or $V := H^1(\Omega)$ together with natural Neumann boundary conditions.

(II) We set $V := H_0^1(\Omega)$. The spatial differential operator has the form $A(u) = -\Delta u + g(t,x,u)$, where the nonlinearity $g : I \times \Omega \times \mathbb{R}$ is measurable with respect to (t,x) for any $u \in \mathbb{R}$ and satisfies the following conditions:

(a) g is continuously differentiable with respect to u for almost all $(t,x) \in I \times \Omega$ and there is a $K > 0$ such that

$$\|g(\cdot,\cdot,0)\|_{L^\infty(I\times\Omega)} + \|\partial_u g(\cdot,\cdot,0)\|_{L^\infty(I\times\Omega)} \le K.$$

(b) The first derivative of g is uniformly Lipschitz-continuous in u on bounded sets, that is, for any $S > 0$ there is $L(S) > 0$ such that

$$\|\partial_u g(\cdot,\cdot,u_1) - \partial_u g(\cdot,\cdot,u_2)\|_{L^\infty(I\times\Omega)} \le L(S)|u_1 - u_2|$$

for any $u_1, u_2 \in \mathbb{R}$ with $|u_1|, |u_2| \le S$.

(c) For almost all $(t,x) \in I \times \Omega$ and all $u \in \mathbb{R}$, the monotonicity condition

$$\partial_u g(\cdot,\cdot,u) \ge 0$$

is fulfilled.

For the data we require the additional regularity $u_0 \in L^\infty(\Omega)$ and $f \in L^q(I \times \Omega)$ where $q > \frac{d}{2} + 1$.

The set of conditions (II) ensures additionally that a solution of the discretized problem can be obtained by Newton's method. For the first case, it is shown for example in [31, Chapter IV, §26] that the problem is well posed, for the second example a proof can be found in [21].

One instance of a non-linearity g that fulfills the set of conditions (II) is a function of the form $g(t,x,u) = h(u)$ with $h : \mathbb{R} \to \mathbb{R}$ a two times continuously differentiable and monotonically increasing map. Our first numerical example (see Section 8.1) is of this type.

3 Galerkin discretization

To discretize the problem (1) in time we define a partition of the half open time interval $\tilde{I} := (0,T]$ into half open subintervals $I_m = (t_{m-1}, t_m]$, with $m = 1, \dots M$ and

$$0 = t_0 < t_1 < \cdots < t_M = T.$$

Furthermore, we denote the length of the subintervals by $k_m = |I_m|$ and the maximal length of the time steps by $k = \max k_m$. The test and trial space for the semidiscretization in time with a discontinuous Galerkin method of order r (dG(r) discretization) is given as

$$X_k^r := \left\{ \Psi \in L^2(I,V) \,\middle|\, \Psi|_{I_m} \in \mathscr{P}_r(I_m, V),\ m = 1, \dots, M \right\},$$

where $\mathscr{P}_r(I_m,V)$ denotes the space of polynomials up to degree r on the interval I_m with values in V. For functions $\Psi \in X_k^r$ we introduce the notations

$$\Psi_m^- := \lim_{\varepsilon \searrow 0} \Psi(t_m - \varepsilon), \quad \Psi_m^+ := \lim_{\varepsilon \searrow 0} \Psi(t_m + \varepsilon) \quad \text{and} \quad [\Psi]_m := \Psi_m^+ - \Psi_m^-.$$

Then the standard formulation of the semidiscrete dG(r) equation reads: Find $u_k \in X_k^r$ such that

$$\sum_{m=1}^{M} (\partial_t u_k, \Phi)_{I_m} + \int_I a(u_k)(\Phi)\,\mathrm{d}t + \sum_{m=0}^{M-1} ([u_k]_m, \Phi_m^+) = (f,\Phi)_I \quad \forall \Phi \in X_k^r, \qquad (2)$$

where $[u_k]_0$ denotes $u_{k,0}^+ - u_0$. The duality pairing in the time derivative term can be replaced by an inner product here since the time derivative of a polynomial in $\mathscr{P}_r(V)$ lies in $\mathscr{P}_{r-1}(V)$.

We assume that the semidiscrete equation has a unique solution, a proof of this fact for the considered class of linear equations satisfying (I) can be found for example in [27, Chapter 12], however in the case of Neumann boundary conditions only the case $\gamma > 0$ is covered. For the considered semilinear example (II) a proof is given in [19].

To obtain a fully discrete equation, next the semidiscrete problem (2) needs to be discretized in space. Therefore we consider shape-regular meshes $\mathscr{T}_h$ on Ω consisting of open quadrilateral or hexahedral cells K, depending on whether the space dimension d is 2 or 3, see, e. g., [5, Chapter 2]. The discretization parameter h for a mesh $\mathscr{T}_h = \{K\}$ denotes the maximal diameter of a cell K. We define a conforming finite element space V_h^s of order s on the mesh $\mathscr{T}_h$ in the usual way via

$$V_h^s := \{ v_h \in V \,|\, v_h|_K \in \mathscr{Q}_s(K) \text{ for } K \in \mathscr{T}_h \},$$

where the polynomial space $\mathscr{Q}_s(K)$ is obtained by bi- or trilinear transformation respectively from the space $\hat{\mathscr{Q}}_s(\hat{K})$ defined on the reference cell $\hat{K} := (0,1)^d$ by

$$\hat{\mathscr{Q}}_s(\hat{K}) := \operatorname{span}\left\{ \prod_{j=1}^{d} x_j^{\alpha_j} \,\middle|\, \alpha_j \in \mathbb{N}_0,\, \alpha_j \le s \right\}.$$

Using the space V_h^s we can define the fully discrete solution space

$$X_{k,h}^{r,s} := \left\{ \Psi \in L^2(I,V) \,\middle|\, \Psi|_{I_m} \in \mathscr{P}_r(I_m, V_h^s),\ m = 1,\dots,M \right\} \subseteq X_k^r.$$

Remark 1 (i) For simplicity, we consider only discretizations based on quadrilateral or hexahedral cells, however note that the presented ideas can be applied on meshes consisting of triangular and tetraedal cells almost without modification. (ii) Here we chose the same spatial trial space V_h^s for all time intervals. But due to the temporal discontinuities of the space X_k^r, a generalization of the described concepts to a spatial discretization varying over time is straightforward. For example, one could choose an individual mesh for each time step, see, e. g., [22].

Since the considered spatial discretization is conforming, the semidiscrete equation (2) translates directly to the fully discrete case. We search for $u_{kh} \in X_{k,h}^{r,s}$ such that

$$\sum_{m=1}^{M} (\partial_t u_{kh}, \Phi)_{I_m} + \int_I a(u_{kh})(\Phi)\,\mathrm{d}t + \sum_{m=0}^{M-1} ([u_{kh}]_m, \Phi_m^+) = (f, \Phi)_I \tag{3}$$

holds true for all $\Phi \in X_{k,h}^{r,s}$.

Due to the discontinuity of the test space at the temporal nodes, we can choose the test functions on each interval independently. By testing with functions that have non-zero values on just a single time interval, we see that (3) is equivalent to the implicit time stepping scheme

$$(\partial_t u_{kh}, \Phi)_{I_m} + \int_{I_m} a(u_{kh})(\Phi)\,\mathrm{d}t + ([u_{kh}]_{m-1}, \Phi_{m-1}^+) = (f, \Phi)_{I_m} \\ \forall \Phi \in \mathscr{P}_r(I_m, V_h^s),\ m = 1, \ldots, M. \tag{4}$$

Let $\{\psi_i \in \mathscr{P}_r(I_m, \mathbb{R}) \,|\, i = 0, \ldots, r\}$ be a basis of the polynomial space $\mathscr{P}_r(I_m, \mathbb{R})$ and $\{\varphi_n \in V_h^s \,|\, n = 1, \ldots, N\}$ be a basis of V_h^s. Then, $\{\psi_i \varphi_n \,|\, i = 0, \ldots, r,\ n = 1, \ldots, N\}$ forms a basis of $\mathscr{P}_r(I_m, V_h^s)$ and the time stepping equation (4) can be rewritten as a system of $(r+1)N$ scalar nonlinear equations for each discretization interval

$$(\partial_t u_{kh}, \psi_i \varphi_n)_{I_m} + \int_{I_m} a(u_{kh})(\varphi_n) \cdot \psi_i\,\mathrm{d}t + \psi_i(t_{n-1})([u_{kh}]_{m-1}, \varphi_n) = (f, \psi_i \varphi_n)_{I_m}, \\ i = 0, \ldots, r,\ n = 1, \ldots, N. \tag{5}$$

4 Solution of the discrete equations by Newton's method

To solve the nonlinear system (5) numerically, we apply Newton's method. Starting from an initial guess for the solution on the current interval, a sequence of approximations is computed by repeatedly solving the Newton update equation. Since we restrict our considerations to the single time interval I_m and in order to keep notations simple, we will denote the Newton iterates by $u_{kh}^l \in \mathscr{P}_r(I_m, V_h^s)$ with the iteration index $l \in \mathbb{N}_0$ without indicating the interval. We define the Newton residuals as

$$R_{i,n}^l := (f, \psi_i \varphi_n)_{I_m} - (\partial_t u_{kh}^l, \psi_i \varphi_n)_{I_m} - \int_{I_m} a(u_{kh}^l)(\varphi_n) \cdot \psi_i\,\mathrm{d}t \\ + \psi_i(t_{m-1})(u_{kh,m-1}^- - u_{kh}^l(t_{m-1}), \varphi_n), \quad i = 0, \ldots, r,\ n = 1, \ldots, N. \tag{6}$$

We denote the partial derivative of the semilinear form a with respect to u evaluated in direction $\delta u \in X$ and tested with $\Phi \in X$ by

$$a_u'(u)(\delta u, \Phi) := \frac{\mathrm{d}}{\mathrm{d}\tau} a(u + \tau \delta u, \Phi)\big|_{\tau=0}.$$

Then the Newton update $w_{kh}^l := u_{kh}^{l+1} - u_{kh}^l$ solves the linear system

$$(\partial_t w_{kh}^l, \psi_i \varphi_n)_{I_m} + \int_{I_m} a_u'(u_{kh}^l)(w_{kh}^l, \psi_i \varphi_n)\,\mathrm{d}t + \psi_i(t_{m-1})(w_{kh}^l(t_{m-1}), \varphi_n) = R_{i,n}^l, \quad i = 0,\dots,r,\ n = 1,\dots,N. \quad (7)$$

Next, we write the Newton update in the chosen temporal and spatial basis, that is $w_{kh}^l = \sum_{j=0}^r \sum_{n'=1}^N W_{j,n'}^l \psi_j \varphi_{n'}$ with real coefficients $W_{j,n'}^l$. Collecting the temporal derivative and jump terms simplifies the update equation to

$$\sum_{j=0}^{r} \sum_{n'=1}^{N} \left[\left(\int_{I_m} \partial_t \psi_j \psi_i \,\mathrm{d}t + \psi_j(t_{m-1})\psi_i(t_{m-1}) \right) (\varphi_{n'}, \varphi_n) + \int_{I_m} a_u'(u_{kh}^l)(\varphi_{n'}, \varphi_n) \cdot \psi_j \psi_i \,\mathrm{d}t \right] W_{j,n'}^l = R_{i,n}^l, \quad i = 0,\dots,r,\ n = 1,\dots,N. \quad (8)$$

For given i and j, evaluating the expression $\int_{I_m} a_u'(u_{kh}^l)(\varphi_{n'}, \varphi_n) \cdot \psi_j \psi_i \,\mathrm{d}t$ for all $n, n' = 0,\dots,N$ corresponds to a weighted temporal integral over the stiffness matrix of the linearized spatial differential operator, which is usually too expensive to be computed numerically. Thus, we approximate it by a suitable mean value

$$\int_{I_m} a_u'(u_{kh}^l)(\varphi_{n'}, \varphi_n) \cdot \psi_j \psi_i \,\mathrm{d}t \approx \overline{a_u'(u_{kh}^l)(\varphi_{n'}, \varphi_n)} \int_{I_m} \psi_j \psi_i \,\mathrm{d}t.$$

Since we perform a Newton iteration even for linear problems and the residual is computed without this approximation, the accuracy of the computed final solution is not affected. Rather, the convergence behaviour of the Newton iteration changes. Introducing the midpoint $\tilde{t}_m := \frac{t_{m-1}+t_m}{2}$ of the current time interval, the most obvious choice for the mean value is

$$\overline{a_u'(u_{kh}^l)(\varphi_{n'}, \varphi_n)} := a_u'(u_{kh}^l(\tilde{t}_m))(\varphi_{n'}, \varphi_n),$$

that is, we evaluate the derivative once at the midpoint of the time interval. For some cases of highly non-linear problems, however, numerical tests indicate a more reliable convergence of the Newton iteration when averaging a_u' over the time interval with a higher order quadrature formula like the two-point Gauss rule.

We introduce the mass matrix $M \in \mathbb{R}^{N\times N}$ and the averaged stiffness matrix $\bar{A} \in \mathbb{R}^{N\times N}$ with the entries

$$M_{n,n'} = (\varphi_{n'}, \varphi_n) \text{ and } \bar{A}_{n,n'} = \overline{a_u'(u_{kh}^l)(\varphi_{n'}, \varphi_n)} \text{ respectively.}$$

With the notations

$$\alpha_{ij} := \int_{I_m} \partial_t \psi_j \psi_i \,\mathrm{d}t + \psi_j(t_{m-1})\psi_i(t_{m-1}) \text{ and } \beta_{ij} = \frac{1}{k_m} \int_{I_m} \psi_j \psi_i \,\mathrm{d}t,$$

the Newton update equation with the approximation for the linearized form discussed above reads

$$\begin{pmatrix} \alpha_{00}M + k_m\beta_{00}\bar{A} & \alpha_{01}M + k_m\beta_{01}\bar{A} & \cdots & \alpha_{0r}M + k_m\beta_{0r}\bar{A} \\ \alpha_{10}M + k_m\beta_{10}\bar{A} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \alpha_{r0}M + k_m\beta_{r0}\bar{A} & \cdots & \cdots & \alpha_{rr}M + k_m\beta_{rr}\bar{A} \end{pmatrix} \begin{pmatrix} W_0^l \\ W_1^l \\ \vdots \\ W_r^l \end{pmatrix} = \begin{pmatrix} R_0^l \\ R_1^l \\ \vdots \\ R_r^l \end{pmatrix}, \quad (9)$$

where the vectors $W_j^l := \begin{pmatrix} W_{j,1}^l & \cdots & W_{j,N}^l \end{pmatrix}^T$ and $R_i^l := \begin{pmatrix} R_{i,1}^l & \cdots & R_{i,N}^l \end{pmatrix}^T$ collect the corrections and residual terms respectively for one temporal basis function.

For fixed r we can give explicit representations for the coefficients α_{ij} and β_{ij} as shown in the following lemma.

Lemma 1 *If we represent the temporal basis $\{\psi_j\}$ as $\psi_j(t) = \sum_{\mu=0}^r c_{\mu j}\left(\frac{t-t_{m-1}}{k_m}\right)^\mu$ and denote the corresponding coefficient matrix by $\mathbf{C} := (c_{\mu j})_{\mu,j\in\{0,\dots,r\}}$, we get the explicit representations*

$$\mathbf{A} = \mathbf{C}^T\mathbf{G}\mathbf{C} \text{ and } \mathbf{B} = \mathbf{C}^T\mathbf{H}\mathbf{C}$$

for the matrices $\mathbf{A} := (\alpha_{ij})_{i,j\in\{0,\dots,r\}}$ and $\mathbf{B} := (\beta_{ij})_{i,j\in\{0,\dots,r\}}$. Here, $\mathbf{H}$ denotes the $(r+1)$-dimensional Hilbert matrix, that is, $\mathbf{H}_{\mu\nu} = \frac{1}{\mu+\nu+1}$, and the entries of $\mathbf{G}$ are given by $\mathbf{G}_{00} = 1$ and $\mathbf{G}_{\mu\nu} = \frac{\nu}{\mu+\nu}$ for the remaining entries.

Proof Transforming the integral $\beta_{ij} = \int_{I_m} \frac{1}{k_m}\psi_j\psi_i\,\mathrm{d}t$ to the unit interval yields

$$\begin{aligned}\beta_{ij} &= \int_0^1 \psi_j(k_m\tau)\psi_i(k_m\tau)\,\mathrm{d}\tau = \sum_{\mu=0}^r\sum_{\nu=0}^r \int_0^1 c_{\nu j}c_{\mu i}\,\tau^{\mu+\nu}\,\mathrm{d}\tau \\ &= \sum_{\mu=0}^r c_{\mu i}\sum_{\nu=0}^r\int_0^1 \tau^{\mu+\nu}\,\mathrm{d}\tau\, c_{\nu j} = \sum_{\mu=0}^r c_{\mu i}\sum_{\nu=0}^r \frac{1}{\mu+\nu+1}\,c_{\nu j}.\end{aligned}$$

Rewriting this identity in terms of matrix products gives the representation formula for $\mathbf{B}$. To derive a representation for $\mathbf{A}$ we start again by transforming the integral to the unit interval and get

$$\begin{aligned}\alpha_{ij} &= \int_0^1 k_m\partial_t\psi_j(k_m\tau)\psi_i(k_m\tau)\,\mathrm{d}\tau + \psi_j(t_{m-1})\psi_i(t_{m-1}) \\ &= \int_0^1 \partial_\tau\psi_j(k_m\tau)\psi_i(k_m\tau)\,\mathrm{d}\tau + c_{0j}c_{0i}.\end{aligned}$$

We proceed as above by plugging in the monomial representation for the temporal basis, evaluating integrals and derivatives of the monomials and rewriting the result in terms of matrix products. □

For order $r = 0$, the resulting scheme is some variant of the well known implicit Euler method, depending on how the temporal integrals in the residual terms are evaluated. This type of scheme is easy to implement on top of existing finite element code for elliptic spatial problems, as long as the matrices M and $\bar{A}$ can be assembled, a linear solver for those matrices is available and elementary vector operations are implemented. If the order r is greater than 0, however, the update equation (9) has higher dimension than the corresponding spatial problem. So a matrix with higher dimension and a different sparsity pattern has to be assembled and solved for. Our goal is now to avoid assembling the full system (9). For this purpose we will approximate it by a scheme consisting of several solution steps of essentially the same form as for the implicit Euler and requiring only matrices and vectors of dimension $N = \dim V_h^s$.

5 Inexact solver for the linear subproblems

The matrix of the equation system (9) for the Newton update is of size $(r+1)^2N^2$ where N is the number of degrees of freedom for the spatial discretization. Hence it is computationally expensive to assemble it completely. In addition depending on the structure of the underlying finite element code for the spatial problem it might not be straightforward to incorporate a matrix of greater dimension than what the spatial discretization suggests. Our approach to work around assembling the whole system matrix is based on block-wise elimination.

To apply block elimination to the update equation (9) it is required that the blocks of the matrix commute, which can be achieved by multiplying all block equations from left with the inverse mass matrix. All blocks of the resulting matrix contain linear combinations of the identity matrix and the matrix $L := k_m M^{-1}\bar{A}$ which makes them commute.

Note that the matrix L is in general not symmetric, but has non-negative real spectrum if the assumptions we made in Section 2 for the operator A are fulfilled. This can be seen by looking at an eigenvalue λ of L and the corresponding eigenvector v. We have $Lv = \lambda v$ which is equivalent to $\bar{A}v = \lambda k_m M v$. Multiplying with the conjugate transpose v^* of v and solving for λ gives $\lambda = \frac{v^*\bar{A}v}{k_m v^* M v}$ and since the right hand side of this identity is a non-negative real value the same holds for the eigenvalue λ.

Block-wise row operations transform the system (9) into a system of the form

$$\begin{pmatrix} p_{00}(L) & p_{01}(L) & \cdots & p_{0r}(L) \\ 0 & p_{11}(L) & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & p_{rr}(L) \end{pmatrix} \begin{pmatrix} W_0^l \\ W_1^l \\ \vdots \\ W_r^l \end{pmatrix} = \sum_{j=0}^{r} \begin{pmatrix} \hat{p}_{0j}(L)R_j^l \\ \hat{p}_{1j}(L)R_j^l \\ \cdots \\ \hat{p}_{rj}(L)R_j^l \end{pmatrix} \tag{10}$$

with the polynomials $\hat{p}_{ij}$ representing the transformations of the right hand side. Since we did not require a particular strategy for the elimination process (like, e. g., LU decomposition or LU decomposition with pivoting), the above system is not uniquely determined for given order r and temporal basis.

The blocks of the upper triangular matrix consist of polynomials p_{ij} of at most degree $i+1$ in the matrix L. The polynomials on the right hand side $\hat{p}_{ij}$ have maximally degree i. However, lower degree can be achieved in some cases by a careful elimination strategy. Obviously it is not feasible to solve equations involving polynomials of order greater than one in L as matrices since taking powers of the matrix $\bar{A}$ will result in a matrix with largely increased stencil and condition number. Therefore those higher order polynomials can only be dealt with efficiently if they can be decomposed into linear factors. If that is possible, solving with them as matrix can be done in several steps with matrices of the form $\mathrm{Id}+\theta L$. To avoid assembling L, the equations for those steps can be multiplied from left by M on both sides.

The obvious question is now whether the temporal basis and the elimination strategy can be chosen in such a way that all polynomials p_{ii} with $i = 0,\dots,r$ have real roots and therefore can be decomposed into linear factors. To gain insight into this issue, we consider a simple homogeneous ordinary differential equation in one dimension,

$$u_t + \lambda u = 0, \quad u(0) = u_0 \tag{11}$$

for some $\lambda > 0$ as test problem.

Since the problem is linear, the Newton iteration reduces to a single step of solving the linear subproblem (9) and if we choose $U^0 = 0$ as starting value, the update W^0 from the first step is already the dG(r) solution of the problem on the current interval represented with respect to the chosen temporal basis.

Next, we need to establish a link between the dG(r) solution of this problem and a Padé approximation of the exact solution $e^{-\lambda t}$.

The $[r/r+1]$-Padé approximation of $e^{-\lambda}$ is a rational function in λ with enumerator of maximal degree r and denominator of maximal degree $r+1$ that approximates $e^{-\lambda}$ up to an error of $\mathscr{O}(\lambda^{2r+2})$. If enumerator and denominator are required to have no common roots, it is uniquely determined (see, e. g. [3]).

Lemma 2 *Let U_1 be an approximate solution of the test problem* (11) *at time $T = 1$ obtained by a single dG(r) time step. Then the identity $U_1 = Q_{r,r+1}(\lambda)u_0$ holds, where $Q_{r,r+1}$ denotes the $[r/r+1]$-Padé approximation of the exponential $e^{-\lambda}$.*

Proof A proof of this statement can be found in [14]. □

Lemma 3 *For any choice of the temporal basis and arbitrary elimination strategy, the denominator of the $[r/r+1]$-Padé approximation of $e^{-\lambda}$ divides the product $\prod_{i=0}^{r} p_{ii}(\lambda)$ of the diagonal elements on the left hand side of* (10).

Proof Fixing a temporal temporal basis $\{\psi_i\}_{i=0}^{r}$, we consider again the approximate solution U_1 of the test problem (11) at time $T = 1$ computed in a single time step. If we start the Newton iteration with $U^0 = 0$, then the residual for the first iterate is given by $R_i^0 = \psi_i(0)u_0$ and after solving the Newton update equation (9) we obtain $U_1 = \sum_{i=0}^{r} \psi_i(1)W_i^0$. Expressing the components W_i^0 of the Newton update by backward substitution in the triangular system (10) gives

$$U_1 = \sum_{i=0}^{r} \psi_i(1)W_i^0 = \sum_{i=0}^{r} \psi_i(1) \frac{\sum_{j=0}^{r} \hat{p}_{ij}(\lambda)\psi_j(0)u_0 - \sum_{j=i+1}^{r} p_{ij}(\lambda)W_j^0}{p_{ii}(\lambda)}.$$

If we substitute the solution components W_j^0 recursively by their explicit representations computed from backward substitution, this results in an expression of the form

$$U_1 = \frac{\kappa(\lambda)}{\tau(\lambda)} u_0,$$

with κ and τ being polynomials with respect to λ. Note that $\tau(\lambda)$ divides $\prod_{i=0}^{r} p_{ii}(\lambda)$. On the other hand, we know from Lemma 2 that U_1 can be written in terms of the $[r/r+1]$-Padé approximation $Q_{r,r+1}$ which gives the identity

$$\frac{\kappa(\lambda)}{\tau(\lambda)} = Q_{r,r+1}(\lambda).$$

It is shown in [20, Chapter V] that the Padé table for $e^{-\lambda}$ is normal and hence the denominator of the $[r/r+1]$-Padé approximation of $e^{-\lambda}$ has exactly degree $r+1$ for any r and no terms in the Padé approximation cancel out. Hence the denominator divides $\tau(\lambda)$, which shows the claim. □

The previous lemma implies that all factors of the denominator of the $[r/r+1]$-Padé approximation occur in the polynomials p_{ii} that show up on the left hand sides of the equations which constitute the decoupled system. Therefore the initial question whether those polynomials can be decomposed into linear factors is related to the question whether the denominator of the Padé approximation can be decomposed into linear factors.

As an immediate consequence from Theorem 8 in [28], this denominator has no more than one simple real root. Thus, for r greater than zero, it is not possible to find a decoupled solution scheme where all polynomials p_{ii} can be decomposed into linear factors over the real numbers.

The same problem was encountered by Schötzau, Schwab and coworkers (see, e. g., [23]) when attempting to decouple the equation system by means of diagonalization. Their solution consists of switching to complex arithmetics. We want to avoid this for two reasons. First the added computational effort for complex calculations is significant, second and more importantly, not for every existing finite element code the implementation of complex equations is straightforward.

The approach we want to take here is to replace polynomials with complex roots by approximations with only real roots. As a consequence the Newton update equation (9) will be solved only approximately. However we will show that under certain conditions the resulting inexact iteration still converges rapidly to the exact solution albeit requiring some more Newton steps.

From Lemma 3 we know that it cannot be circumvented to approximate the denominator of the $[r/r+1]$-Padé approximation. Therefore we attempt to avoid further terms with complex roots in the diagonal polynomials p_{ii}. For the last polynomial p_{rr} such terms cannot occur as the following lemma shows.

Lemma 4 *For any choice of the temporal basis and arbitrary elimination strategy, the polynomial p_{rr} divides the denominator of the $[r/r+1]$-Padé approximation.*

Proof Considering again the test problem (11), an equation for W_r^0 can be derived alternatively by applying Cramer's rule on (9). Denoting the columns of the matrices $\mathbf{A}$ and $\mathbf{B}$ as defined in Lemma 1 by $\mathbf{A}_j$ and $\mathbf{B}_j$ respectively and the vector of the residuals by R^0 we obtain

$$\det(\mathbf{A}+\lambda\mathbf{B})W_r^0 = \det\left(\mathbf{A}_0+\lambda\mathbf{B}_0, \mathbf{A}_1+\lambda\mathbf{B}_1, \ldots, \mathbf{A}_{r-1}+\lambda\mathbf{B}_{r-1}, R^0\right). \tag{12}$$

Since the polynomial degree of p_{rr} is less or equal $r+1$ and the solution W_r^0 is uniquely determined for each λ, p_{rr} divides $\det(\mathbf{A}+\lambda\mathbf{B})$. For the polynomial $\det(\mathbf{A}+\lambda\mathbf{B})$, we obtain with Lemma 1

$$\det(\mathbf{A}+L\mathbf{B}) = \det(\mathbf{C}^T)\det(\mathbf{G}+L\mathbf{H})\det(\mathbf{C}),$$

and therefore except for multiplication by a scalar constant the same polynomial for any choice of the temporal basis. This polynomial has to be the denominator of the $[r/r+1]$-Padé approximation. To see this, we choose a Lagrange basis with the last node at the end of the time interval. In that case, we have $W_r^0 = U_1$ and Lemma 2 shows the claim. □

From now on we only consider bases where p_{rr} is a scalar multiple of the denominator of the Padé approximation. This has two reasons: on the one hand without further structural insight it is difficult to find a basis for which p_{rr} is of lower degree than $r+1$ and on the other hand this allows for a more efficient approximation of the factors with complex roots as we shall see.

Next we answer the question how to approximate the denominators of the Padé approximation in order to avoid complex roots. Since the spatial differential operator was chosen in such a way that the spectrum of the matrix L is real and positive, it is sufficient if the approximation is of good quality for positive real numbers. This motivates measuring the approximation quality of a given approximation $\tilde{p}_{rr}$ by the quantity

$$\gamma_r := \sup\left\{\left|1-\tfrac{p_{rr}(\lambda)}{\tilde{p}_{rr}(\lambda)}\right| \mid \lambda \in \mathbb{R}_0^+\right\},$$

which gives the maximal relative error for positive real numbers.

The polynomials p_{rr} for $r \leq 6$, together with possible approximations $\tilde{p}_{rr}$ and corresponding quality values γ_r are listed in Table 1. The approximations were chosen to be exact for $\lambda = 0$ and asymptotically exact for $\lambda \to \infty$, which fixes the first and last coefficient. Approximating by a polynomial consisting of identical linear factors turned out to yield smaller γ_r than decomposing p into quadratic factors and approximating those individually. As an additional benefit, the resulting numerical algorithm uses the same matrix for each of the intermediate solution steps, which reduces computing time.

So instead of computing the exact Newton update W_r^l, we use the approximation $\widetilde{W}_r^l$ given by

$$\tilde{p}_{rr}(L)\widetilde{W}_r^l = \sum_{j=0}^{r} \hat{p}_{rj}(L)R_j^l. \tag{13}$$

Table 1 Polynomials p_{rr} and their approximations for different values of r

r	$p_{rr}(\lambda)$	$\tilde{p}_{rr}(\lambda)$	γ_r
1	$1+\frac{2}{3}\lambda+\frac{1}{6}\lambda^2$	$(1+\sqrt{\frac{1}{6}}\lambda)^2$	0.092
2	$1+\frac{3}{5}\lambda+\frac{3}{20}\lambda^2+\frac{1}{60}\lambda^3$	$(1+\sqrt[3]{\frac{1}{60}}\lambda)^3$	0.169
3	$1+\frac{4}{7}\lambda+\frac{1}{7}\lambda^2+\frac{2}{105}\lambda^3+\frac{1}{840}\lambda^4$	$(1+\sqrt[4]{\frac{1}{840}}\lambda)^4$	0.238
4	$1+\frac{5}{9}\lambda+\frac{5}{36}\lambda^2+\frac{5}{252}\lambda^3+\frac{5}{3024}\lambda^4+\frac{1}{15120}\lambda^5$	$(1+\sqrt[5]{\frac{1}{15120}}\lambda)^5$	0.301
5	$1+\frac{6}{11}\lambda+\frac{3}{22}\lambda^2+\frac{2}{99}\lambda^3+\frac{1}{528}\lambda^4+\frac{1}{9240}\lambda^5$ $+\frac{1}{332640}\lambda^6$	$(1+\sqrt[6]{\frac{1}{332640}}\lambda)^6$	0.359
6	$1+\frac{7}{13}\lambda+\frac{7}{52}\lambda^2+\frac{35}{1716}\lambda^3+\frac{7}{3432}\lambda^4+\frac{7}{51480}\lambda^5$ $+\frac{7}{1235520}\lambda^6+\frac{1}{8648640}\lambda^7$	$(1+\sqrt[7]{\frac{1}{8648640}}\lambda)^7$	0.412

This equation can be solved in $r+1$ solution steps with the linear factors of $\tilde{p}_{rr}(L)$ as matrix. To obtain approximations $\widetilde{W}_0^l,\dots,\widetilde{W}_{r-1}^l$ for the other solution components, we perform backward substitution with $\widetilde{W}_r^l$ in (10). To ensure that this works without requiring powers of the matrix L and therefore further approximations, it is necessary that the polynomials p_{ii} for $i=1,\dots,r-1$ can be decomposed into linear factors over $\mathbb{R}$. Moreover if the roots of the linear factors were contained in the spectrum of L, the matrix we have to solve for would become singular. To rule out this possibility all roots of the polynomials p_{ii} should be negative. From our computations we know that this can be achieved by judicious choice of the temporal basis $\{\psi_j\}$ and a suitable elimination strategy at least up to order $r=4$. However it is not known if complex and non-negative roots can be avoided for any r.

In total, for a given triangular system (10) and an approximation $\tilde{p}_{rr}$ of the bottom right polynomial, the approximate Newton update is the solution of

$$\begin{pmatrix} p_{00}(L) & p_{01}(L) & \cdots & p_{0r}(L) \\ 0 & p_{11}(L) & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \tilde{p}_{rr}(L) \end{pmatrix} \begin{pmatrix} \widetilde{W}_0^l \\ \widetilde{W}_1^l \\ \vdots \\ \widetilde{W}_r^l \end{pmatrix} = \sum_{j=0}^{r} \begin{pmatrix} \hat{p}_{0j}(L)R_j^l \\ \hat{p}_{1j}(L)R_j^l \\ \cdots \\ \hat{p}_{rj}(L)R_j^l \end{pmatrix}. \tag{14}$$

If we use the presented approximation scheme for the Newton update equation, the resulting inexact Newton iteration takes the form of Algorithm 1.

6 Convergence of the inexact Newton iteration

The proposed solution scheme for the Newton update equation can be seen as an inexact Newton iteration involving three sources of inexactness: averaging the matrix $\bar{A}$ over the time interval introduced an error into the Jacobian, another inexactness was introduced by the approximation of p_{rr} by $\tilde{p}_{rr}$ proposed in the previous section and finally since the dimension of the resulting linear systems is large, they have to

Algorithm 1

Require: Starting value $\widetilde{U}^0$
for $l = 0, 1, 2, \ldots$ **do**
  Compute residuals $R_0^l, \ldots, R_r^l$ from (6)
  if stopping criterion fulfilled **then**
    break
  end if
  Compute approximate last update component $\widetilde{W}_r^l$ from last row of (14)
  Compute remaining update components $\widetilde{W}_0^l, \ldots, \widetilde{W}_{r-1}^l$ by backward substitution in (14)
  Set $\widetilde{U}^{l+1} := \widetilde{U}^l + \widetilde{W}^l$
end for

be solved iteratively. In our numerical realization we employ a multigrid solver with an incomplete LU decomposition as smoother for this purpose.

The latter error can be neglected here since we require a low relative tolerance for the linear solver. That means the multigrid iteration is only terminated when the error is smaller than what the termination criterion for the Newton iteration allows.

To assess the inexactness introduced by averaging the stiffness matrix, we need a priori knowledge about the non-linearity under consideration. Since this error can be reduced if necessary by choosing a smaller time step and vanishes for the linear case, we wont analyse it either. This seems justified because most existing implicit Runge-Kutta implementations perform a similar approximation by evaluating the Jacobian only at the initial value and strategies exist to counter non-converging steps by adaptively reducing the size of the time step (see for example [11]).

Instead we focus on the influence of replacing p_{rr} by $\tilde{p}_{rr}$. The impact of the proposed approximation for the linear sub-problems on the convergence of the Newton iteration can be analysed in the context of inexact Newton methods. Such methods are characterized by the fact that they do not compute the exact Newton update W^l but only an approximation $\widetilde{W}^l$. Convergence results are typically given in terms of the relative error of the approximation. The following lemma gives an estimate for the relative error of the proposed approximated Newton update.

Lemma 5 *For a fixed approximation scheme, let $\widetilde{W}^l$ be the solution of* (14) *and W^l be the solution of the exact update equation* (10)*. Moreover, let the rational functions q_j, $j = 0, \ldots, r$ be defined recursively by*

$$q_r(\lambda) := 1 - [\tilde{p}_{rr}(\lambda)]^{-1} p_{rr}(\lambda),$$
$$q_i(\lambda) := [p_{ii}(\lambda)]^{-1} \sum_{j=i+1}^{r} p_{ij}(\lambda) q_j(\lambda) \quad \text{for } i = 0, \ldots, r-1.$$

We assume that all roots of the polynomials p_{ii} are negative for all $i = 0, \ldots, r-1$. Then for the relative error of the approximated Newton update $\widetilde{W}^l$, the estimate

$$\|W^l - \widetilde{W}^l\|_2^2 \leq \nu_r \|W^l\|_2^2$$

holds with

$$\nu_r := \sum_{i=0}^{r} \sup \left\{ q_i(\lambda)^2 \,\middle|\, \lambda \in \mathbb{R}_0^+ \right\}.$$

Proof We split the error into its components

$$\|W^l - \widetilde{W}^l\|_2^2 = \sum_{i=0}^{r} \|W_i^l - \widetilde{W}_i^l\|_2^2$$

and start by deriving an explicit representation for the last component of the error in terms of the exact update. From (10) and (14) we obtain

$$\tilde{p}_{rr}(L)\widetilde{W}_r^l = \bar{R}_r = p_{rr}(L)W_r^l,$$

and therefore

$$W_r^l - \widetilde{W}_r^l = \left(\mathrm{Id} - [\tilde{p}_{rr}(L)]^{-1} p_{rr}(L)\right) W_r^l = q_r(L)W_r^l.$$

For the other components, the block upper triangular system (10) gives by means of backward substitution

$$W_i^l - \widetilde{W}_i^l = [p_{ii}(L)]^{-1} \left[\sum_{j=i+1}^{r} p_{ij}(L) \left[\widetilde{W}_j^l - W_j^l\right] \right].$$

An induction argument results in the representation

$$W_i^l - \widetilde{W}_i^l = q_i(L)W_r^l$$

for i in $\{0,\dots,r-1\}$. In terms of those explicit representations, the error of the approximated Newton update reads

$$\|W^l - \widetilde{W}^l\|_2^2 = \sum_{i=0}^{r} \|q_i(L)W_r^l\|_2^2 \leq \|W_r^l\|_2^2 \sum_{i=0}^{r} \|q_i(L)\|_2^2 \leq \|W^l\|_2^2 \sum_{i=0}^{r} \|q_i(L)\|_2^2.$$

Note that especially for large r the last step can be a considerable overestimation of the true error. Computing the matrix norms on the right hand side with analytic functional calculus (see for example [6, Chapter VII]) gives

$$\|W^l - \widetilde{W}^l\|_2^2 \leq \|W^l\|_2^2 \sum_{i=0}^{r} \sup\left\{ q_i(\lambda)^2 \,\middle|\, \lambda \in \sigma(L) \right\},$$

where $\sigma(L)$ denotes the spectrum of L. Functional calculus can be applied here since due to the assumption that all roots of the diagonal polynomials p_{ii} for $i = 0,\dots,r-1$ are negative, the rational functions q_i are analytic in a neighbourhood of $\sigma(L)$. Recalling that $\sigma(L) \subseteq \mathbb{R}_0^+$, the assertion is shown. □

A good approximate linear solution scheme for a dG(r) method should attempt to minimize the estimate from Lemma 5 for all matrices L with positive and real spectrum. A lower bound for the best possible estimate is given by the quality value γ_r of the approximating polynomial $\tilde{p}_{rr}$. Besides that, the choice of the temporal basis and the strategy employed for the block elimination have a significant impact on the estimate.

Since the estimate for the relative error of the Newton update is independent of the iteration index l and in particular not decreasing for large l, we cannot expect to prove super-linear convergence for the inexact Newton iteration. For linear convergence we quote the following special case of a theorem given in [32].

Theorem 1 *Let $F : D \subseteq \mathbb{R}^d \to \mathbb{R}^d$ vanish at $\hat{U} \in \mathbb{R}^d$, that is, $F(\hat{U}) = 0$. We further assume there exists $\rho > 0$ such that F is Fréchet differentiable in $B_\rho(\hat{U})$ with invertible Jacobian and fulfils*

$$\mu := \sup \left\{ \|DF(\hat{U})^{-1}(DF(V) - DF(W))\|_2 \,\middle|\, V \neq W; V, W \in B_\rho(\hat{U}) \right\} < \infty.$$

Moreover, let $\nu < 1$, $U^0 \in B_\rho(\hat{U})$ and $U^{l+1} = U^l + \widetilde{W}^l$ for $l \in \mathbb{N}$ with the update $\tilde{W}^l$ satisfying

$$\frac{\|\widetilde{W}^l - W^l\|_2}{\|W^l\|_2} \leq \nu,$$

where W^l is the exact Newton update at U^l given by

$$DF(U^l)W^l = -F(U^l).$$

Then the sequence $\{U^l\}$ of the inexact Newton iterates satisfies

$$\|U^{l+1} - \hat{U}\|_2 \leq \eta \|U^l - \hat{U}\|_2, \text{ with } \eta = \nu + \frac{(1+\nu)\mu}{1-\mu}. \tag{15}$$

To apply this theorem to our solution scheme, we choose F to map a nodal representation vector U of a discrete function u_{kh} to the corresponding residual vector. Since μ can be made arbitrarily small by decreasing ρ, this shows local linear convergence almost with the rate ν_r for an approximate solution scheme of the presented form if the relative error ν_r of the approximate update $\widetilde{W}^l$ can be shown to be less than one by means of Lemma 5. That means if the starting value for the Newton iteration is sufficiently close to the solution on the current time step, the Newton iteration will converge linearly with rate almost ν_r to the solution. Since we start the iteration with the value at the end of the last time step, this can be ensured if the time step is sufficiently short.

7 Realizations for first to third order

In this section we present concrete realizations of simplified solution schemes of first, second and third order constructed according to the described principles. It turned out that for those orders all requirements for a well-working scheme discussed in the previous sections can be met when using Lagrange polynomials with equally spaced nodes including the end points of the time interval as temporal basis. This allows for an immediate interpretation of the computed solution components as nodal values.

Due to the involved formulas the complete schemes for dG(2) and dG(3) are to be found in the appendix.

7.1 Scheme for dG(1)

For the dG(1) scheme with Lagrange basis on the time interval the update equation (9) with the coefficients α_{ij} and β_{ij} from Lemma 1 reads

$$\begin{pmatrix} \frac{1}{2}\mathrm{Id}+\frac{1}{3}L & \frac{1}{2}\mathrm{Id}+\frac{1}{6}L \\ -\frac{1}{2}\mathrm{Id}+\frac{1}{6}L & \frac{1}{2}\mathrm{Id}+\frac{1}{3}L \end{pmatrix} \begin{pmatrix} W_0^l \\ W_1^l \end{pmatrix} = \begin{pmatrix} M^{-1}R_0^l \\ M^{-1}R_1^l \end{pmatrix}. \tag{16}$$

Note that the equation was scaled block-wise with the inverse mass matrix to allow for block elimination. Solving for W_1^l gives the expression

$$\left(\mathrm{Id}+\frac{2}{3}L+\frac{1}{6}L^2\right) W_1^l = \left(\mathrm{Id}-\frac{1}{3}L\right) M^{-1}R_0^l + \left(\mathrm{Id}+\frac{2}{3}L\right) M^{-1}R_1^l.$$

We approximate the polynomial on the left hand side by $\left(\mathrm{Id}+\frac{1}{\sqrt{6}}L\right)^2$ (see Table 1). To allow for reuse of this matrix, we derive the equation for $\widetilde{W}_0^l$ by adding the first row of (16) multiplied by $\sqrt{\frac{2}{3}}+\frac{2}{3}$ to the second row multiplied by $\sqrt{\frac{2}{3}}-\frac{4}{3}$. Then we have $p_{00}(\lambda) = 1+\frac{1}{\sqrt{6}}\lambda$ and get

$$\left(\mathrm{Id}+\frac{1}{\sqrt{6}}L\right)\widetilde{W}_0^l = \frac{2+\sqrt{6}}{3}M^{-1}R_0^l + \frac{\sqrt{6}-4}{3}M^{-1}R_1^l - \left(\frac{\sqrt{6}-1}{3}\mathrm{Id}+\frac{\sqrt{6}-2}{6}L\right)\widetilde{W}_1^l$$

Computing $\widetilde{W}_1^l$ can be split into two steps introducing a temporary variable $\widetilde{V}_1^l$. Since the matrix $L = k_m M^{-1}\bar{A}$ cannot be assembled efficiently, we multiply the three equations of the resulting solution scheme by M and obtain the computable scheme

$$\begin{aligned}
&\left(M+\sqrt{\frac{1}{6}}k_m\bar{A}\right)\widetilde{V}_1^l = R_0^l + R_1^l + \frac{1}{3}k_m\bar{A}M^{-1}\left(2R_1^l - R_0^l\right), \\
&\left(M+\sqrt{\frac{1}{6}}k_m\bar{A}\right)\widetilde{W}_1^l = M\widetilde{V}_1^l, \\
&\left(M+\sqrt{\frac{1}{6}}k_m\bar{A}\right)\widetilde{W}_0^l = \frac{2+\sqrt{6}}{3}R_0^l + \frac{\sqrt{6}-4}{3}R_1^l - \left(\frac{\sqrt{6}-1}{3}M+\frac{\sqrt{6}-2}{6}k_m\bar{A}\right)\widetilde{W}_1^l.
\end{aligned}$$

Evaluating this scheme involves three linear solution steps with the same matrix $M+\sqrt{\frac{1}{6}}k_m\bar{A}$. Note that each of the solution steps has the same basic structure as an implicit Euler time step. In addition to that we have to solve one time for the mass matrix when assembling the right hand side for the first equation. However, solving for the mass matrix will typically be quite cheap since on uniformly refined grids its condition number is approximately one which allows for efficient solution for example with the conjugate gradient method. For non-uniform grids like the ones resulting from adaptive refinement, a Jacobi preconditioner will give comparable efficiency, see for example [29]. To assess the contraction rate we can expect from this scheme, we estimate the rational functions q_0 and q_1 as defined in Lemma 5 and obtain

$$|q_0(\lambda)| \leq 0.0321 \quad \text{and} \quad |q_1(\lambda)| \leq \gamma_1 = 0.092 \quad \text{for } \lambda \in \mathbb{R}^+.$$

Hence, Lemma 5 yields the estimate

$$\|\widetilde{W}^l - W^l\|_2 \leq 0.097\|W^l\|_2.$$

7.2 Scheme for dG(2)

For the dG(2) method, again using the Langrange basis, the coefficient matrices evaluate to

$$\mathbf{A} = \begin{pmatrix} \frac{1}{2} & \frac{2}{3} & -\frac{1}{6} \\ -\frac{2}{3} & 0 & \frac{2}{3} \\ \frac{1}{6} & -\frac{2}{3} & \frac{1}{2} \end{pmatrix} \text{ and } \qquad \mathbf{B} = \begin{pmatrix} \frac{2}{15} & \frac{1}{15} & -\frac{1}{30} \\ \frac{1}{15} & \frac{8}{15} & \frac{1}{15} \\ -\frac{1}{30} & \frac{1}{15} & \frac{2}{15} \end{pmatrix}.$$

As in the dG(1) case, we solve (9) (after block-wise scaling with M^{-1}) for W_2^l and apply the polynomial approximation from Table 1 on the left hand side to get for $\widetilde{W}_2^l$

$$\left(\mathrm{Id} + \sqrt[3]{\frac{1}{60}}L\right)^3 \widetilde{W}_2^l = G^l$$

with some right hand side G^l depending on the residuals. After decomposition into three steps, this time using two temporary values $\widetilde{V}_1^l$ and $\widetilde{V}_2^l$ and subsequent scaling with the mass matrix, we obtain a computable scheme for $\widetilde{W}_2$ which is given in Equation (19) in the appendix.

If we attempt to solve for $\widetilde{W}_1^l$ next, we get $p_{11}(L) = \mathrm{Id} - \frac{2}{5}L$. This matrix can become singular if L has the Eigenvalue $\frac{5}{2}$, which puts a severe restriction on the range of valid step sizes and prevents the application of Lemma 5. However, first solving for $\widetilde{W}_0^l$ gives a matrix that does not suffer from this problem. Formally, this is equivalent to a permutation of the Lagrange basis. For computing $\widetilde{W}_1^l$ we reuse the left hand side that was used for $\widetilde{W}_0^l$. This results in the two solution steps shown in Equation (20) in the appendix.

Concerning the computational effort of this scheme, we note that for computing $\widetilde{W}_2^l$, three solution steps with identical matrices are carried out. Computing the other two components involves two inversions of one additional matrix. So in total five linear equation systems involving two different system matrices have to be solved. Additionally, two inversions of the mass matrix are required for assembling the right hand side.

For this dG(2) scheme, the rational functions q_i defined in Lemma 5 can be estimated by

$$|q_0(\lambda)| \leq 0.085, \quad |q_1(\lambda)| \leq 0.161 \quad \text{and} \quad |q_2(\lambda)| \leq \gamma_2 = 0.169 \quad \text{for } \lambda \in \mathbb{R}^+.$$

This gives the upper bound

$$\|\widetilde{W}^l - W^l\|_2 \leq 0.248\|W^l\|_2$$

for the contraction estimate .

7.3 Scheme for dG(3)

For dG(3), we start once more by computing the last component $\widetilde{W}_3$ with the proposed approximate scheme using the polynomial from Table 1, which results in the scheme with four solution steps given in Equation (21) in the appendix.

Computing the next solution component leads to a quadratic polynomial in L on the left hand side. It turns out that when solving for $\widetilde{W}_0^l$ first, this polynomial has two negative roots, that is, we can decompose it into linear factors and get an equation of the form

$$\left(\mathrm{Id}+\frac{2}{13+\sqrt{29}}L\right)\left(\mathrm{Id}+\frac{2}{13-\sqrt{29}}L\right)\widetilde{W}_0^l = G_0^l$$

which can be solved in two steps.

The remaining two components are computed in such a way that we can reuse the matrix on the left hand side of the equation for the second step of computing $\widetilde{W}_0^l$. The complete scheme for computing $\tilde{W}_0^l$, $\tilde{W}_1^l$ and $\tilde{W}_2^l$ is given in Equation (22) of the appendix.

One Newton step with this scheme requires the solution of eight linear equation systems with three different system matrices and four inversions of the mass matrix for assembling the right hand side, whereas the full scheme would use a matrix with sixteen times larger stencil than that of each system matrix. The rational functions q_i for the dG(3) scheme can be estimated by

$$|q_0(\lambda)| \le 0.084,\ |q_1(\lambda)| \le 0.033,\ |q_2(\lambda)| \le 0.016,\ \text{and}\ |q_3(\lambda)| \le \gamma_3 = 0.238$$

for $\lambda \in \mathbb{R}^+$. Hence the total contraction estimate according to Lemma 5 reads

$$\|\widetilde{W}^l - W^l\|_2 \le 0.256\|W^l\|_2.$$

8 Numerical results

In this section we demonstrate the advantages of the proposed evaluation schemes by numerical experiments on two examples. First we consider a simple semilinear equation in two space dimensions (see Section 8.1). As a second test case, a model for a combustion process is studied in Section 8.2. All computations were carried out using the finite element library Gascoigne [25] and some subroutines from the optimization package RoDoBo [26].

8.1 Example 1: Semilinear Test Problem

As a first test-case we study a simple semilinear equation. For the spatial domain we choose the two-dimensional unit square $\Omega = (0,1)^2$ and consider the equation

$$\begin{aligned} \partial_t u - \Delta u + u^3 &= f \quad \text{in } I\times\Omega, \\ u|_{\partial\Omega} &= 0 \quad \text{on } I\times\partial\Omega, \\ u(0) &= 0 \end{aligned} \tag{17}$$

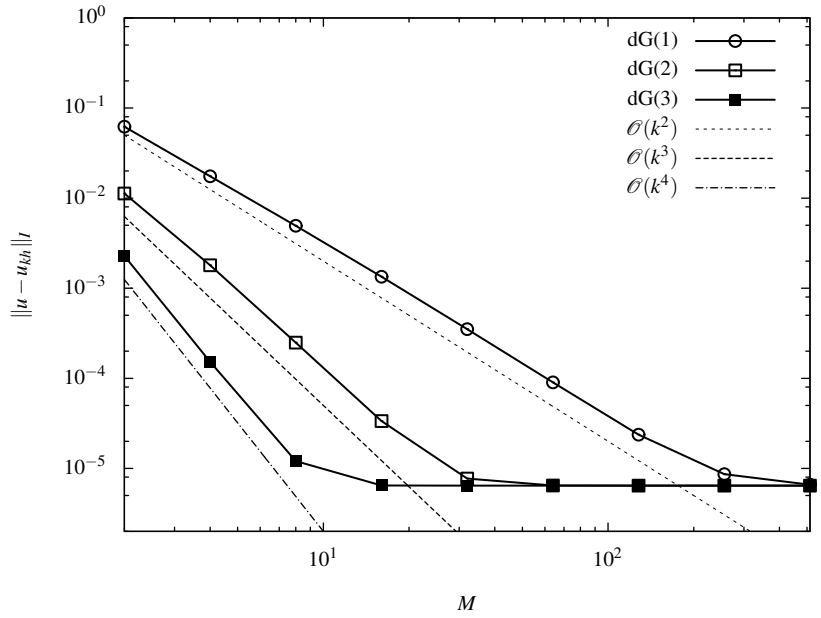


Fig. 1 *Example 1:* Discretization error $\|u-u_{kh}\|_I$ for space discretization with 4096 biquadratic elements

on the time interval $I=(0,\frac{1}{2})$. The right hand side is given by

$$f(t,x_1,x_2)=\sin(\pi x_1)\sin(\pi x_2)\Big\{\pi^2\cos(\pi^2 t)+10 \\ +\big(\sin(\pi^2 t)+10t\big)\Big[2\pi^2+\big(\sin(\pi^2 t)+10t\big)^2\sin(\pi x_1)^2\sin(\pi x_2)^2\Big]\Big\}.$$

This problem has the exact solution

$$u(t,x_1,x_2)=\big(\sin(\pi^2 t)+10t\big)\sin(\pi x_1)\sin(\pi x_2)$$

and it can be seen easily that the set of assumptions (II) is satisfied.

To analyse the error of the temporal dG(r) semidiscretization for different r and different step sizes k we use a fixed spatial mesh consisting of 4096 equally-sized quadrilaterals. The space discretization is done with biquadratic Lagrange elements and for the time discretization we consider a sequence of uniformly refined meshes with equidistant nodes, that is, we have $k=1/M$. Figure 1 shows the behaviour of the discretization error with respect to the L^2 norm on $I\times\Omega$ for orders $r=1,2,3$. The convergence order $\mathcal{O}(k^{r+1})$ of the dG(r) method can be observed in all three cases until reaching the spatial accuracy.

Figure 2 shows the accuracy of the computed solution at final time T, measured with respect to the $L^2(\Omega)$ norm. For the dG(1) and dG(2) method, nodal superconvergence of third and fifth order respectively at the last time node is observed clearly. This shows that nodal superconvergence with order $2r+1$, which was proved for

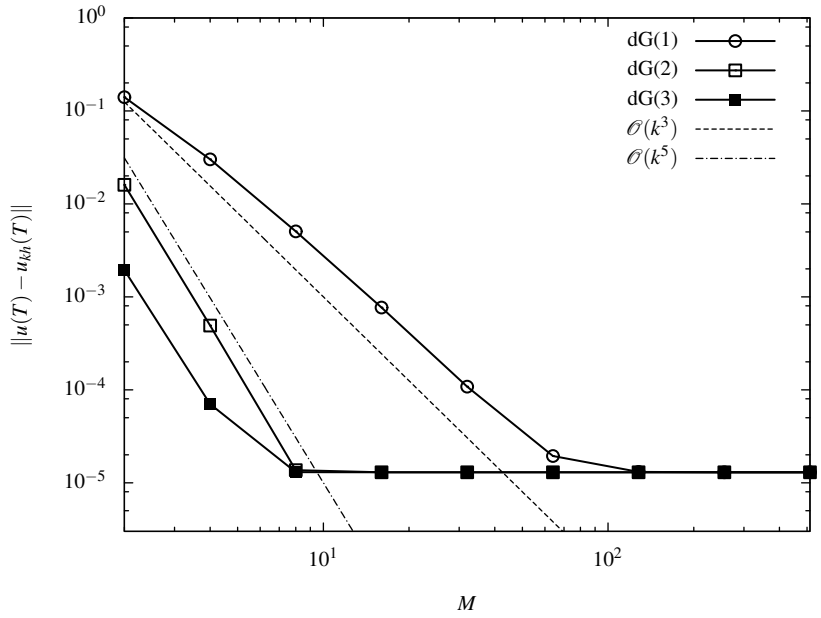


Fig. 2 *Example 1:* Discretization error $\|u(T)-u_{kh}(T)\|$ at final time T for space discretization with 4096 biquadratic elements

example in [27, Chapter 12] for linear equations, occurs also for this nonlinear problem. For the dG(3) method, tests with an ODE version of the test equation reveal that superconvergence occurs only for very short time steps, therefore in the PDE case it cannot be observed due to the dominance of the spatial discretization error. We assume that the missing superconvergence for larger time steps can be attributed to order reduction phenomenons similar to those discussed in [10].

To evaluate the computational cost of the proposed simplified dG(r) scheme, we compare its run time to the run time of solving the full Newton equation (7). For the full Newton update, we consider on the one hand solving the update equation (9), that is, averaging the Jacobian of the partial differential operator over the time interval, and on the other hand assembling the blocks of the Jacobian exactly by integrating the derivatives of the semilinear form with an $r+1$ point Gauss formula on each time interval. Since the sizes of the involved matrices and vectors are different for the full schemes and the simplified approach, it seems not reasonable to compare counts of matrix and vector operations. Instead we implemented all three approaches on top of the same finite element code while restricting the overhead to a minimum in order to obtain comparable CPU times.

In all three cases, we recompute the Jacobian not in every iteration, but only when the contraction rate with respect to the residual becomes too large. This is a common practice to improve the speed of implicit single step methods. Since for exact Newton's method we expect fast superlinear convergence, the matrix will be kept for the next step only if the contraction rate is no worse than 0.01. For our simplified

iteration, this setting is not reasonable since our convergence estimate predicts the rate of convergence to be around ν_r. Hence we choose the threshold $0.01 + \nu_r$ for the simplified scheme. The full Newton scheme with averaged Jacobian is an inexact Newton's method as well, therefore we use the same thresholds $0.01 + \nu_r$.

For the dG(1) method we obtain comparable run times for the exact Jacobian and for our simplified scheme, as seen in Figure 3(a). Therefore the results for the averaged Jacobian are omitted. The performance of the second and third order methods can be seen from Figure 3(b) and Figure 3(c). Except for very small time steps, the simplified solution schemes provide a considerable speed improvement over both variants involving a full-sized Jacobian. From the gap between the exact and the averaged Jacobian we conclude that the computational costs of integrating the Jacobian with respect to time clearly outweigh the gains from quadratic convergence of the nonlinear solver. This provides a justification for averaging it in each time interval.

In Figure 3(d) we compare how computing time relates to the obtained accuracy for the simplified schemes of first to third order. Although the required computations are getting considerably more involved and the contraction rate of the approximate Newton iteration drops when increasing the order, using higher orders pays off for the test example since the number of time steps to reach a desired accuracy is greatly reduced.

8.2 Combustion problem

The equation for this test problem gives a simplified model for a gaseous combustion process and is taken from [13, Chapter VII, §2]. We consider a simple one-species reaction. Under the low Mach number hypothesis and with further approximations the motion of the fluid becomes independent from temperature and concentration of the species. With a prescribed solenoidal velocity field we can separately solve the equations for concentration and temperature of the species. Here we will assume the velocity to be zero.

We denote the species concentration by Y and introduce the dimensionless temperature

$$\theta = \frac{T - T_{\text{unburnt}}}{T_{\text{burnt}} - T_{\text{unburnt}}}$$

Then the combustion process is modelled by the two equations

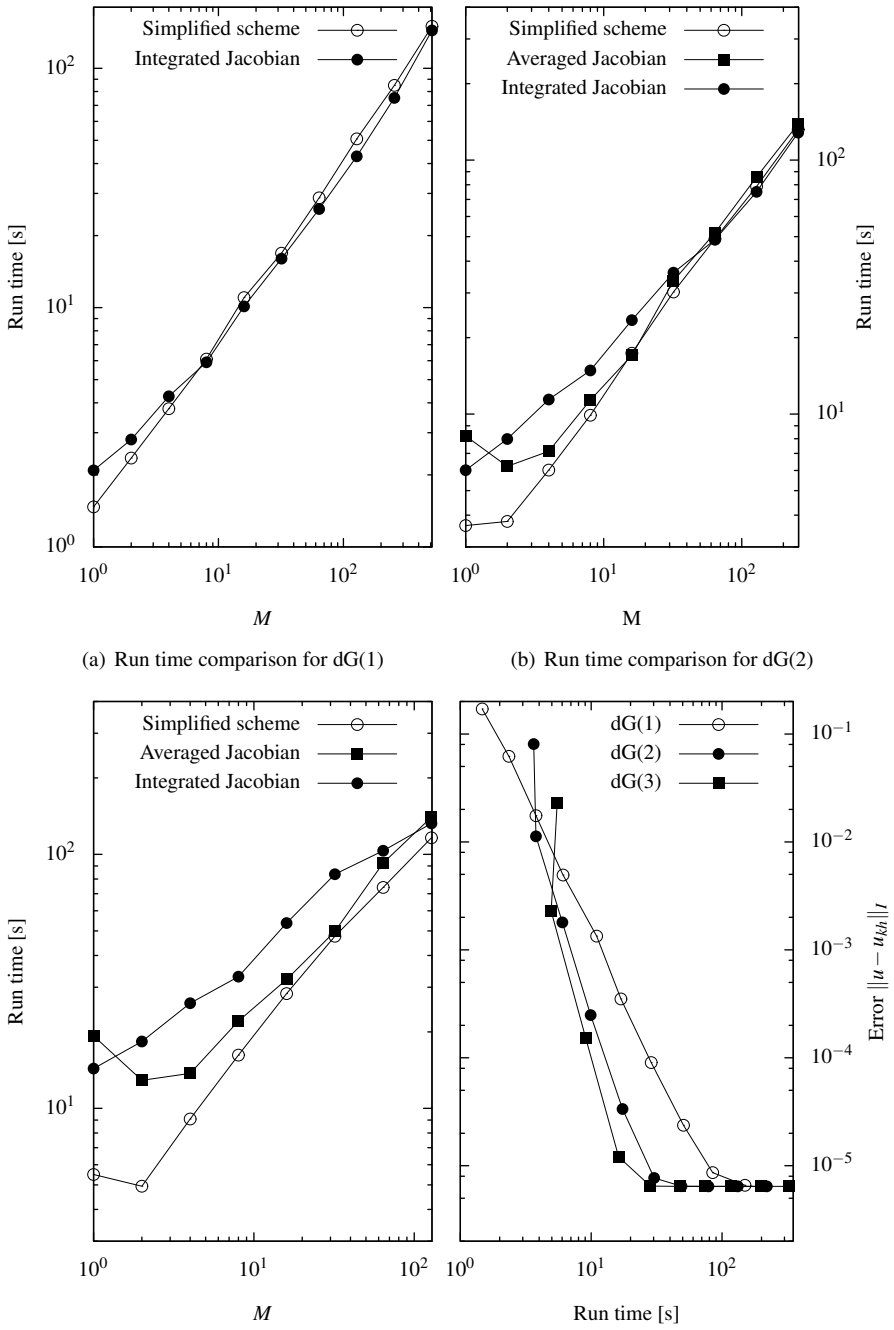
$$\partial_t \theta - \Delta \theta = \omega(Y, \theta), \tag{18a}$$

$$\partial_t Y - \frac{1}{\text{Le}} \Delta Y = -\omega(Y, \theta), \tag{18b}$$

where the Lewis Number Le is the ratio of diffusivity of heat and diffusivity of mass. The reaction mechanism is a simple one-species mechanism governed by an Arrhenius law

$$\omega(Y, \theta) = \frac{\beta^2}{2\text{Le}} Y e^{\frac{\beta(\theta-1)}{1+\alpha(\theta-1)}}$$

in which an approximation for large activation energy has been employed. Here, we consider a freely propagating laminar flame in two space dimensions and its response



(a) Run time comparison for dG(1)

(b) Run time comparison for dG(2)

(c) Run time comparison for dG(3)

(d) Relationship between run time and global accuracy of the solution with respect to the L^2 norm for the reduced schemes

Fig. 3 *Example 1:* Performance comparison of different solution schemes

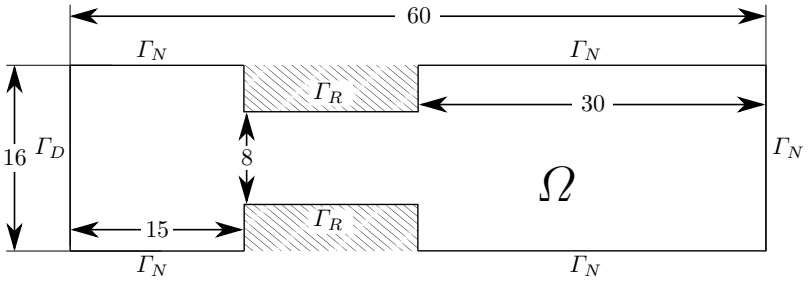


Fig. 4 *Example 2:* Spatial domain Ω for the combustion problem

to a heat absorbing obstacle, a set of cooled parallel rods with rectangular cross section as seen in Figure 4. The boundary conditions are chosen as

$$\begin{aligned}
\theta &= 1 \quad \text{on } \Gamma_D \times (0,T),\\
Y &= 0 \quad \text{on } \Gamma_D \times (0,T),\\
\partial_n \theta &= 0 \quad \text{on } \Gamma_N \times (0,T),\\
\partial_n Y &= 0 \quad \text{on } \Gamma_N \times (0,T),\\
\partial_n \theta + k\theta &= 0 \quad \text{on } \Gamma_R \times (0,T),\\
\partial_n Y &= 0 \quad \text{on } \Gamma_R \times (0,T),
\end{aligned}$$

were the Robin boundary condition on Γ_R models the heat absorption of the rods.

To complete the description of this test problem, we indicate the initial condition as the analytical solution of a one-dimensional right-traveling flame in the limit $\beta \to \infty$ located left of the obstacle

$$\begin{aligned}
\theta(0,x) &= \begin{cases} 1 & \text{for } x_1 \le \tilde{x}_1 \\ e^{\tilde{x}_1 - x_1} & \text{for } x_1 > \tilde{x}_1 \end{cases} \text{ on } \Omega,\\
Y(0,x) &= \begin{cases} 0 & \text{for } x_1 \le \tilde{x}_1 \\ 1 - e^{\mathrm{Le}(\tilde{x}_1 - x_1)} & \text{for } x_1 > \tilde{x}_1 \end{cases} \text{ on } \Omega,
\end{aligned}$$

together with the set of parameters

$$\mathrm{Le} = 1, \quad \alpha = 0.8, \quad \beta = 10, \quad k = 0.1, \quad \tilde{x}_1 = 9.$$

The well-posedness of the problem can be shown by standard arguments employing Brouwer's fixed point theorem. Regarding our assumptions on the derivative of the spatial part of the differential operator, we note that it is obviously self-adjoint. Numerical evaluation of the spectrum of the resulting stiffness matrices after discretization suggests that it is also positive.

We simulate the propagating flame on the time interval $(0,60)$. The physical quantity we are interested in is the mean reaction rate over the whole simulated time. It is

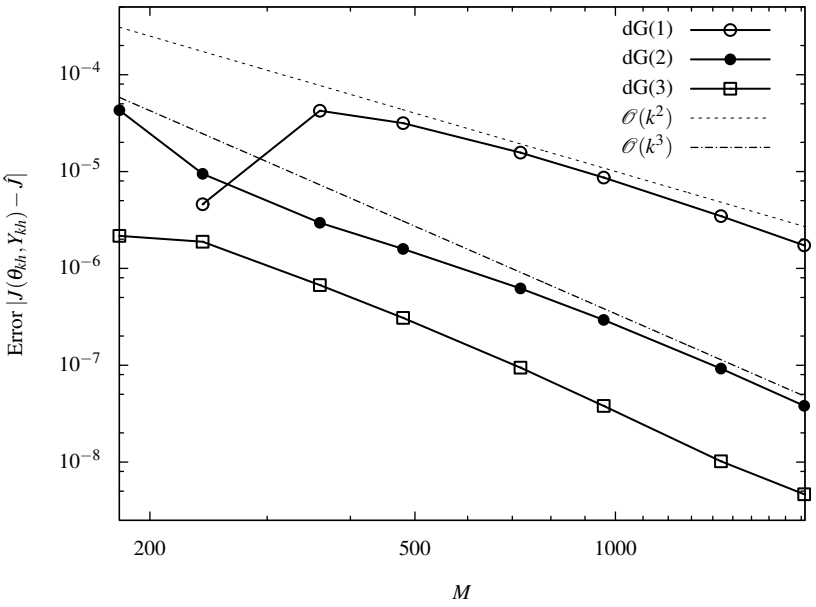


Fig. 5 *Example 2:* Error $|J(\theta_{kh}, Y_{kh}) - \hat{J}|$ of the dG solution with order 1 to 3 for uniform time discretization with M time steps

given by

$$J(\theta, Y) = \frac{1}{60|\Omega|} \int_0^{60} \int_\Omega \omega(\theta, Y) \, \mathrm{d}x \, \mathrm{d}t.$$

To get an idea what rate of convergence to expect for this functional, we consider the weak formulation of the temperature equation (18a) and test with the constant function with value 1. Then, apart from a scalar factor, the right hand side becomes the functional J. On the left hand side, the spatial operator vanishes and we are left with integrals over the time derivative of the temperature and over boundary terms. The integral over the time derivative amounts to the value of the temperature at the final time, therefore this term should expose superconvergence. For the boundary terms, however, we only expect them to converge with order $r+1$. So in total for the functional J convergence with at most order $r+1$ can be expected but it will be limited by the low regularity of the initial data and the spatial domain Ω.

For the spatial discretization of the problem, we consider again a uniform mesh and biquadratic Lagrange elements. Since we are only interested in the convergence of the time discretization, we fix the fineness of the spatial discretization at $N = 11041$ nodes. Then a reference value $\hat{J}$ for the functional $J(\theta, Y)$ is obtained by extrapolating the results of the dG(3) method on a sequence of uniform refinements of the time steps on this spatial grid.

Figure 5 shows the convergence of the functional value $J(\theta, Y)$ on a sequence of uniformly refined temporal grids for dG methods of first to third order. As elaborated above, given sufficient regularity of the exact solution (θ, Y), we expect the order of

Table 2 *Example 2:* Relative speed improvement of the simplified solution scheme over an exact Newton method and a Newton method with averaged Jacobian respectively

	dG(1)	dG(2)		dG(3)	
M	S_{exact}	S_{exact}	S_{avg}	S_{exact}	S_{avg}
240	1.0	4.7	3.4	9.0	6.6
480	1.8	5.0	2.6	8.4	3.9
960	1.8	3.9	1.8	6.8	1.9
1920	1.6	2.1	1.2	4.4	1.3

convergence $r+1$ for a dG(r) method. For dG(1) and dG(2), second and third order convergence are observed. Presumably due to insufficient regularity of the solution, the error of the dG(3) result decreases only with third order with respect to the time step size.

Since the nonlinearity in this example is assumed to have a greater influence on the convergence of the Newton method than for the scalar test problem, we adjust the threshold on the contraction rate for reassembling the Jacobian to a higher value. For the exact Newton the matrix is reassembled whenever the contraction rate exceeds 0.03, for all other schemes the threshold is set to $0.03 + \nu_r$. According to our observation, the exact Newton method will benefit most from this adjustment, nevertheless the run time gap to the simplified schemes is even bigger than for the scalar test equation (see Table 2). This is due to the fact that we deal with a system of two equations here.

We measure for every combination of order r and fineness M of the temporal discretization the CPU time t_{simple} consumed by the simplified solution scheme and the CPU time t_{exact} for the exact Jacobian. For the second and third order scheme, we measure in addition the time t_{avg} used by the full system with averaged Jacobian. In Table 2 we list the quotients $S_{exact} := \frac{t_{exact}}{t_{simple}}$ and $S_{avg} := \frac{t_{avg}}{t_{simple}}$ which measure the relative speed improvement of the simplified scheme over the exact Newton and the scheme involving the averaged Jacobian respectively.

We observe a significant performance improvement — up to a reduction of computing time by a factor of nine for dG(3) and $M = 240$ time steps — of our simplified scheme compared to the exact Newton iteration, particularly for high orders and long time steps. The improvement over the averaged Jacobian is not as great but especially for coarse temporal grids, which are of most practical relevance when using high order schemes, it is still significant. In Figure 6 we compare the simplified schemes for first to third order with respect to the relationship between obtained accuracy in the quantity of interest and consumed time. As expected, due to the observed higher order of convergence, the dG(2) method is more efficient than dG(1) for the flame propagation example. Although the dG(3) scheme suffers from order reduction, it takes about the same time to obtain a given accuracy as the dG(2) method since it has a better error constant.

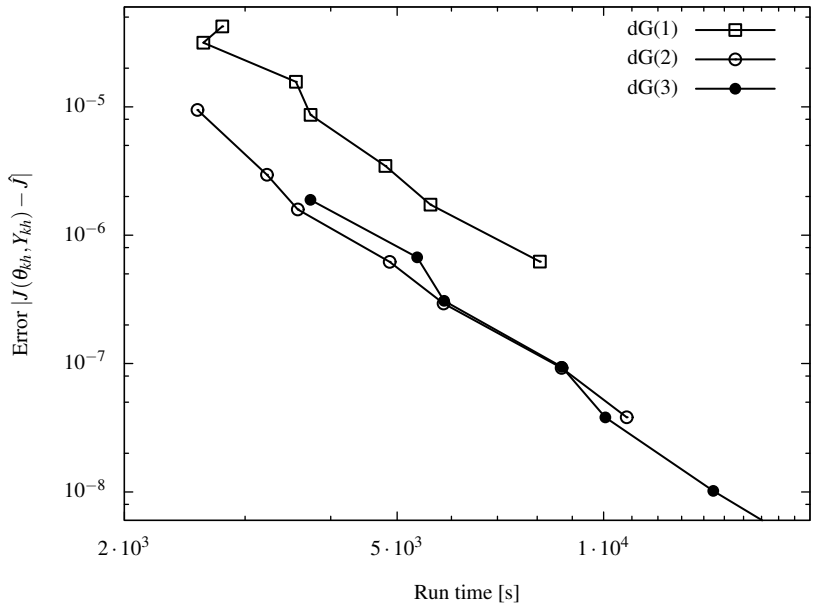


Fig. 6 *Example 2:* Relationship between computing time and obtained accuracy for simplified schemes of first to third order

A Complete time stepping schemes for dG(2) and dG(3)

A.1 Scheme for dG(2)

First, the solution component $\widetilde{W}_2^l$ is computed by the three linear solution steps

$$\begin{aligned}
&\left(M + \sqrt[3]{\frac{1}{60}}\, k_m \bar{A}\right) \widetilde{V}_1^l = R_0^l + R_1^l + R_2^l + \frac{1}{10} k_m \bar{A} M^{-1} \left(-4R_0^l + R_1^l + 6R_2^l\right) \\
&\qquad\qquad + \frac{1}{40} k_m^2 \bar{A} M^{-1} \bar{A} M^{-1} \left(2R_0^l - R_1^l + 6R_2^l\right), \\
&\left(M + \sqrt[3]{\frac{1}{60}}\, k_m \bar{A}\right) \widetilde{V}_2^l = M \widetilde{V}_1^l, \\
&\left(M + \sqrt[3]{\frac{1}{60}}\, k_m \bar{A}\right) \widetilde{W}_2^l = M \widetilde{V}_2^l,
\end{aligned} \tag{19}$$

where $\widetilde{V}_1^l$ and $\widetilde{V}_2^l$ are temporary values. For the other two components, backward substitution gives the linear equations

$$\begin{aligned}
&\left(M + \frac{1}{10} k_m \bar{A}\right) \widetilde{W}_0^l = \frac{6}{5} R_0^l - \frac{3}{10} R_1^l + \frac{6}{5} R_2^l - \frac{1}{5} \left(M + \frac{1}{2} k_m \bar{A}\right) \widetilde{W}_2^l, \\
&\left(M + \frac{1}{10} k_m \bar{A}\right) \widetilde{W}_1^l = \frac{3}{8} R_1^l - \frac{3}{2} R_2^l + 5M \left(\widetilde{W}_0^l + \widetilde{W}_2^l\right) + k_m \bar{A} \left(-\frac{3}{4} \widetilde{W}_0^l + \frac{7}{4} \widetilde{W}_2^l\right).
\end{aligned} \tag{20}$$

A.2 Scheme for dG(3)

To obtain the solution component $\widetilde{W}_3^l$, we solve the four linear equations

$$\begin{aligned}
&\left(M+\sqrt[4]{\frac{1}{840}}k_m\bar{A}\right)\widetilde{V}_1^l = R_0^l+R_1^l+R_2^l+R_3^l \\
&\qquad\qquad +\frac{1}{21}k_m\bar{A}M^{-1}\left(-9R_0^l-2R_1^l+5R_2^l+12R_3^l\right) \\
&\qquad\qquad +\frac{1}{126}\left(k_m\bar{A}M^{-1}\right)^2\left(9R_0^l-2R_1^l+R_2^l+18R_3^l\right) \\
&\qquad\qquad +\frac{1}{5670}\left(k_m\bar{A}M^{-1}\right)^3\left(-27R_0^l+8R_1^l-17R_2^l+108R_3^l\right), \\
&\left(M+\sqrt[4]{\frac{1}{840}}k_m\bar{A}\right)\widetilde{V}_2^l = M\widetilde{V}_1^l, \\
&\left(M+\sqrt[4]{\frac{1}{840}}k_m\bar{A}\right)\widetilde{V}_3^l = M\widetilde{V}_2^l, \\
&\left(M+\sqrt[4]{\frac{1}{840}}k_m\bar{A}\right)\widetilde{W}_3^l = M\widetilde{V}_3^l,
\end{aligned} \tag{21}$$

with temporary variables $\widetilde{V}_1^l$, $\widetilde{V}_2^l$ and $\widetilde{V}_3^l$. The remaining components are given by

$$\begin{aligned}
&\left(M+\frac{2}{13+\sqrt{29}}k_m\bar{A}\right)\widetilde{V}_4^l = G_0^l, \quad \left(M+\frac{2}{13-\sqrt{29}}k_m\bar{A}\right)\widetilde{W}_0^l = M\widetilde{V}_4^l, \\
&\left(M+\frac{2}{13-\sqrt{29}}k_m\bar{A}\right)\widetilde{W}_1^l = G_1^l, \quad \left(M+\frac{2}{13-\sqrt{29}}k_m\bar{A}\right)\widetilde{W}_2^l = G_2^l
\end{aligned} \tag{22}$$

with another temporary value $\widetilde{V}_4^l$ and the right hand sides

$$\begin{aligned}
G_0^l &:= \frac{26}{35}R_0^l-\frac{628}{945}R_1^l+\frac{142}{945}R_2^l-\frac{44}{35}R_3^l \\
&\quad +k_m\bar{A}M^{-1}\left(\frac{18}{35}R_0^l-\frac{4}{45}R_1^l+\frac{22}{315}R_2^l-\frac{12}{35}R_3^l\right) \\
&\quad +\frac{9}{35}M\widetilde{W}_3^l+k_m\bar{A}\left(\frac{4}{35}\widetilde{W}_3^l+\frac{1}{70}k_mM^{-1}\bar{A}\widetilde{W}_3^l\right), \\
G_1^l &:= \frac{1844-52\sqrt{29}}{945}R_0^l+\frac{4568+1256\sqrt{29}}{25515}R_1^l \\
&\quad +\frac{5548-284\sqrt{29}}{25515}R_2^l+\frac{-536+88\sqrt{29}}{945}R_3^l \\
&\quad +M\left(\frac{-26+2\sqrt{29}}{27}\widetilde{W}_0^l+\frac{11-18\sqrt{29}}{945}\widetilde{W}_3^l\right) \\
&\quad +k_m\bar{A}\left(-\frac{4}{27}\widetilde{W}_0^l+\frac{23-9\sqrt{29}}{1890}W_3^l\right),
\end{aligned}$$

$$
\begin{aligned}
G_2^l := &\left(\frac{116}{11025} - \frac{2\sqrt{29}}{1225}\right) R_0^l + \left(\frac{116252}{297675} + \frac{1168\sqrt{29}}{99225}\right) R_1^l \\
&+ \left(\frac{196822}{297675} + \frac{3548\sqrt{29}}{99225}\right) R_2^l + \left(-\frac{9404}{11025} + \frac{64\sqrt{29}}{3675}\right) R_2^l \\
&+ M\left(\frac{-257-9\sqrt{29}}{280}\widetilde{W}_1^l + \frac{-6317+2691\sqrt{29}}{88200}\widetilde{W}_3^l\right) \\
&+ k_m\bar{A}\left(\frac{1349+23\sqrt{29}}{9800}\widetilde{W}_1^l + \frac{-3023+279\sqrt{29}}{88200}\widetilde{W}_3^l\right).
\end{aligned}
$$

Acknowledgements The second author gratefully acknowledges financial support from the Munich Centre of Advanced Computing and the International Graduate School of Science and Engineering at the Technische Universität München.

References

1. Becker, R., Meidner, D., Vexler, B.: Efficient numerical solution of parabolic optimization problems by finite element methods. Optim. Methods Softw. **22**(5), 813–833 (2007)
2. Besier, M., Rannacher, R.: Goal-oriented space-time adaptivity in the finite element galerkin method for the computation of nonstationary incompressible flow. Internat. J. Numer. Methods Fluids (2012). DOI 10.1002/fld.2735
3. Brezinski, C., Van Iseghem, J.: A taste of Padé approximation. In: Acta numerica, 1995, Acta Numer., pp. 53–103. Cambridge Univ. Press, Cambridge (1995)
4. Chrysafinos, K.: Discontinuous galerkin approximations for distributed optimal control problems constrained by parabolic PDEs. Int. J. Numer. Anal. Model. **4**(3-4), 690–712 (2007)
5. Ciarlet, P.G.: The finite element method for elliptic problems, *Classics in Applied Mathematics*, vol. 40. SIAM, Philadelphia, PA (2002)
6. Dunford, N., Schwartz, J.T.: Linear operators. Part I. Wiley Classics Library. John Wiley & Sons Inc., New York (1988)
7. Eriksson, K., Johnson, C.: Adaptive finite element methods for parabolic problems I: A linear model problem. SIAM J. Numer. Anal. **28**(1), 43–77 (1991)
8. Eriksson, K., Johnson, C.: Adaptive finite element methods for parabolic problems II: Optimal error estimates in $L_\infty L_2$ and $L_\infty L_\infty$. SIAM J. Numer. Anal. **32**(3), 706–740 (1995)
9. Eriksson, K., Johnson, C., Thomée, V.: Time discretization of parabolic problems by the discontinuous Galerkin method. M2AN Math. Model. Numer. Anal. **19**, 611–643 (1985)
10. Frank, R., Schneid, J., Überhuber, C.W.: Order results for implicit Runge-Kutta methods applied to stiff systems. SIAM J. Numer. Anal. **22**(3), 515–534 (1985)
11. Hairer, E., Wanner, G.: Stiff differential equations solved by Radau methods. J. Comput. Appl. Math. **111**(1-2), 93–111 (1999)
12. Hussain, S., Schieweck, F., Turek, S.: Higher order Galerkin time discretizations and fast multigrid solvers for the heat equation. J. Numer. Math. **19**(1), 41–62 (2011)
13. Lang, J.: Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems, *Lec. Notes Comput. Sci. Eng.*, vol. 16. Springer-Verlag (2001)
14. Lesaint, P., Raviart, P.A.: On a finite element method for solving the neutron transport equation. In: Mathematical aspects of finite elements in partial differential equations, pp. 89–123. Publication No. 33. Math. Res. Center, Univ. of Wisconsin-Madison, Academic Press, New York (1974)
15. Meidner, D., Vexler, B.: Adaptive space-time finite element methods for parabolic optimization problems. SIAM J. Control Optim. **46**(1), 116–142 (2007)
16. Meidner, D., Vexler, B.: A priori error estimates for space-time finite element approximation of parabolic optimal control problems. Part I: Problems without control constraints. SIAM J. Control Optim. **47**(3), 1150–1177 (2008)
17. Meidner, D., Vexler, B.: A priori error estimates for space-time finite element approximation of parabolic optimal control problems. Part II: Problems with control constraints. SIAM J. Control Optim. **47**(3), 1301–1329 (2008)

18. Meidner, D., Vexler, B.: A Priori Error Analysis of the Petrov–Galerkin Crank–Nicolson Scheme for Parabolic Optimal Control Problems. SIAM J. Control Optim. **49**(5), 2183–2211 (2011)
19. Neitzel, I., Vexler, B.: A priori error estimates for space–time finite element discretization of semilinear parabolic optimal control problems. Numerische Mathematik **120**(2), 345–386 (2011)
20. Perron, O.: Die Lehre von den Kettenbrüchen. Band II. Analytisch-funktionentheoretische Kettenbrüche, 3rd edn. B. G. Teubner Verlagsgesellschaft, Stuttgart (1957)
21. Raymond, J.P., Zidani, H.: Hamiltonian Pontryagin's principles for control problems governed by semilinear parabolic equations. Appl. Math. Optim. **39**(2), 143–177 (1999)
22. Schmich, M., Vexler, B.: Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. SIAM J. Sci. Comput. **30**(1), 369–393 (2008)
23. Schötzau, D., Schwab, C.: Time discretization of parabolic problems by the *hp*-version of the discontinuous Galerkin finite element method. SIAM J. Numer. Anal. **38**(3), 837–875 (2000)
24. Schötzau, D., Wihler, T.: A posteriori error estimation for hp-version time-stepping methods for parabolic partial differential equations. Numer. Math. **115**, 475–509 (2010)
25. GASCOIGNE: The finite element toolkit. http://www.gascoigne.uni-hd.de
26. RODOBO: A C++ library for optimization with stationary and nonstationary PDEs with interface to GASCOIGNE [25]. http://www.rodobo.uni-hd.de
27. Thomée, V.: Galerkin Finite Element Methods for Parabolic Problems, *Spinger Ser. Comput. Math.*, vol. 25. Springer, Berlin (2006)
28. Wanner, G., Hairer, E., Nørsett, S.P.: Order stars and stability theorems. BIT **18**(4), 475–489 (1978)
29. Wathen, A.J.: Realistic eigenvalue bounds for the galerkin mass matrix. IMA J. Numer. Anal. **7**(4), 449–457 (1987)
30. Werder, T., Gerdes, K., Schötzau, D., Schwab, C.: *hp*-discontinuous Galerkin time stepping for parabolic problems. Comput. Methods Appl. Mech. Engrg. **190**(49-50), 6685–6708 (2001)
31. Wloka, J.: Partial differential equations. Cambridge University Press, Cambridge (1987). Translated from the German by C. B. Thomas and M. J. Thomas
32. Ypma, T.J.: Local Convergence of Inexact Newton Methods. SIAM J. Numer. Anal. **21**(3), 583–590 (1984)