hp Discontinuous Galerkin Time Stepping For Parabolic Problems

T. Werder [a,1] K. Gerdes [b] D. Schötzau [c,2] C. Schwab [d,*]

[a] *Inst. of Computational Sciences, ETH, CH-8092 Zürich, Switzerland*

[b] *Dept. of Mathematics, Chalmers University, SE-412 96 Gothenburg, Sweden*

[c] *School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA*

[d] *Seminar für Angewandte Mathematik, ETH, CH-8092 Zürich, Switzerland*

Abstract

The algorithmic pattern of the *hp* Discontinuous Galerkin Finite Element Method (DGFEM) for the time semidiscretization of abstract parabolic evolution equations is presented. In combination with a continuous *hp* discretization in space we obtain a fully discrete *hp*-scheme for the numerical solution of parabolic problems. Numerical examples for the heat equation in a two dimensional domain confirm the exponential convergence rates which are predicted by theoretical results, under realistic assumptions on the initial data and the forcing terms. We also compare different methods to reduce the computational cost of the DGFEM.

Key words: Discontinuous Galerkin Methods, *hp* Finite Element Methods

1 Introduction

Parabolic evolution equations appear in numerous engineering applications such as fluid dynamics or heat transfer. The nature of such problems is transient and, therefore, an appropriate time stepping scheme has to be applied in numerical simulations to obtain an approximative solution. A flexible and efficient time discretization method is the Discontinuous Galerkin Finite Element Method (DGFEM) which is based on variational formulations of initial

[*] Corresponding author, email: `schwab@sam.math.ethz.ch`.

[1] The work of this author was carried out while visiting the Chalmers University whose kind support is greatfully acknowledged.

[2] The work of this author was supported by the Swiss National Science Foundation (Schweizerischer Nationalfonds).

value problems, but still is closely related to implicit Runge-Kutta schemes. Discontinuous Galerkin methods have been introduced in the seventies, and the work by Lesaint and Raviart [21] seems to contain the first error analysis of the DG time stepping method for ODEs. More recently, the DGFEM has been applied to parabolic problems and was studied in a series of papers by Eriksson, Johnson, Thomée and their coworkers [11–18,23]. We also refer to the recent monograph [35] and the references therein. However, in these works the convergence of the discrete solution to the exact one is achieved by reducing the mesh sizes k and h in time and space, respectively, i.e., by letting $k \rightarrow 0$ and $h \rightarrow 0$. Evidently, the convergence mechanism and analysis of the DGFEM time discretization are similar to the ones encountered in the so-called h-version of the Finite Element Method (FEM). By this h-version approach it is only possible to achieve algebraic convergence orders in time and space, see, e.g., [35].

In the early eighties, the p- and hp-version of the FEM appeared in the literature (see, e.g., [2–4,8,9,19,20,26,34] and the references therein). In these approaches higher approximation orders are employed and for linear elliptic problems it has been shown that they lead to arbitrarily high algebraic convergence rates for smooth solutions and even to exponential convergence, provided that the exact solution of the problem is piecewise analytic. A survey of these results can be found in [33] and the references therein. The hp-versions of discontinuous Galerkin methods have already been applied successfully for the spatial discretization of convection-diffusion problems, we refer to [5–7] and the references therein.

Typically, solutions to parabolic problems also are piecewise analytic in time and exhibit time singularities due to incompatible or discontinuous data which are, however, strongly smoothened out in time. Hence, this solution behavior suggests that p- and hp-version concepts can be applied in time discretization methods as well. Attempts in this direction have been made in [1] where the p- and hp-version of continuous and discontinuous Galerkin methods have been investigated. However, severe restrictions on the space discretization limit the applicability of the results there, since the highly anisotropic spatial meshes which are mandatory for the resolution of boundary layers, fronts or corner singularities were not manageable. In the recent work [30,31] it has been shown that the hp-DGFEM is able to resolve time singularities at exponential convergence rates, independently of the spatial discretization, and the predicted exponential convergence in time and space for parabolic equations has been confirmed for one dimensional model problems. We also mention [32] where new a-priori estimates have been derived for the hp-DGFEM for nonlinear initial value problems.

In this work we describe in detail algorithmic and implementational aspects of the hp-DGFEM time stepping for parabolic problems. We pay special attention to efficiency aspects such as the decoupling of the systems in time within every time step. The algorithmic description is done for general parabolic problems, but for our numerical results we restrict ourselves to the heat equation in two space dimensions. The computations for this model problem, with different initial and boundary conditions, confirm the predicted theoretical results on exponential convergence. Furthermore, CPU time comparisons for the h-, p- and hp-DGFEM clearly show the superiority of the hp-version DGFEM in achieving a low error tolerance at minimal costs. We demonstrate the power of the hp-version DGFEM time stepping in conjunction with an hp-FEM in space (again at the example of the heat equation) on an L-shaped domain with a temporal and a spatial singularity. Both singularities are resolved at exponential convergence rates.

The outline of this work is as follows: In Section 2 the DGFEM time discretization for parabolic problems is presented. In Section 3 some a priori error estimates for the DGFEM are collected from [30]. Particularly, we show that, even in the presence of incompatible initial data, exponential convergence can be achieved by the hp-DGFEM. The implementational aspects of the DGFEM are discussed in Section 4 for a fully discrete hp scheme. Section 5 contains a short discussion of possible parallelization strategies. In Section 6 we present numerical results and CPU time measurements obtained for the heat equation in two space dimensions. Conclusions can be found in Section 7.

We use the following standard notation: For a bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$, $d \geq 1$, we write $L^2(\Omega)$ for the usual Lebesgue space with inner product $(\cdot,\cdot)_{L^2(\Omega)}$. By $H^k(\Omega)$, $k \in \mathbb{N}_0$, we denote the Sobolev space with norm $\|u\|_{H^k(\Omega)} = (\sum_{|\alpha|\leq k} \|D^\alpha u\|^2_{L^2(\Omega)})^{\frac{1}{2}}$ and inner product $(u,v)_{H^k(\Omega)} = \sum_{|\alpha|\leq k}(D^\alpha u, D^\alpha v)_{L^2(\Omega)}$, where standard multi-index notation is used. For non-integer indices s, the Sobolev spaces $H^s(\Omega)$ are defined by the K-method of interpolation [22]. $H^1_0(\Omega)$ is the subspace consisting of all $H^1(\Omega)$-functions whose restriction to the boundary $\partial\Omega$ is zero, in the sense of the trace. To describe time discretizations we use Bochner spaces of functions which map a (time) interval $I = (a,b)$ into a Banach space X. $L^2(I;X)$ and $H^s(I;X)$ are the corresponding Lebesgue and Sobolev spaces. $\mathcal{P}^r(I;X)$ denotes the set of all polynomials of degree $\leq r$ with coefficients in X. $C^\infty_0(I;X)$ is the space of all functions $\varphi \in C^\infty(I;X)$ with compact support in the interval I. We omit to write the dependence on X for $X = \mathbb{R}$. $C_b(I;X)$ denotes the bounded continuous functions on the interval I.

2 DGFEM Time Stepping for Parabolic Problems

In this section we introduce the DGFEM time stepping method for parabolic problems in the absence of spatial discretization. To this end, we use the same semidiscrete setting as in [30,31]. A fully discrete scheme will then be considered in Section 4.

2.1 Abstract Parabolic Problems

We consider linear parabolic problems of the form

$$\begin{aligned} u'(t) + Lu(t) &= g(t), \qquad t \in J = (0,T), \\ u(0) &= u_0, \end{aligned} \tag{1}$$

where L is assumed to be an elliptic spatial operator, u_0 the initial datum and g the forcing term. The precise functional framework we use is as follows.

Let X and H be complex and separable Hilbert spaces with dense and compact embedding $X \overset{d}{\hookrightarrow} H$. The norms in X and H are denoted by $\|\cdot\|_X$ and $\|\cdot\|_H$, respectively. The inner product in H is $(\cdot,\cdot)_H$. We assume that the operator L is given as $(Lu,v)_{X^*\times X} = a(u,v)$, where X^* is the dual space of X with corresponding duality pairing $(\cdot,\cdot)_{X^*\times X}$ and $a : X \times X \to \mathbb{R}$ is a continuous, coercive sesquilinear form satisfying

$$\begin{aligned} |a(u,v)| &\le \alpha \|u\|_X \|v\|_X \qquad && \forall u,v \in X, \\ \mathrm{Re}\, a(u,u) &\ge \beta \|u\|_X^2 && \forall u \in X, \\ a(u,v) &= \overline{a(v,u)} && \forall u,v \in X. \end{aligned}$$

It is always assumed that the initial datum satisfies $u_0 \in H$, and that the forcing term satisfies $g \in L^2(J;H)$. The weak formulation of the problem (1) is obtained by multiplication with test functions $v \in X$ and $\varphi(t) \in C_0^\infty(J)$, followed by an appropriate integration by parts. It reads as follows: Find $u \in L^2(J;X) \cap H^1(J;X^*)$ (which implies $u \in C([0,T];H)$) such that $u(0) = u_0$ in H and

$$-\int_J (u(t),v)_H \varphi'(t)dt + \int_J a(u,v)\varphi(t)dt = \int_J (g(t),v)_{X^*\times X}\varphi(t)dt$$

for all $v \in X$ and $\varphi \in C_0^\infty(J)$. For $u_0 \in H$ and $g \in L^2(J;H)$ and under the above assumptions there exists a unique weak solution, see, e.g., [22].

We emphasize here that the spaces X and H typically are infinite dimensional function spaces and, in this sense, the theoretical setting of this section is semidiscrete. In practice, the spatial operator L might also have to be discretized. This will be adressed in Section 4 below.

Example 1 *The standard example of a parabolic equation fitting into the above framework is the heat equation which describes the temperature field $u(x,t)$ in the isotropic body $\Omega \subset \mathbb{R}^d$ (with Dirichlet boundary conditions, i.e.,*

$u|_{\partial\Omega} = 0$) and the time interval $J = (0, T)$ with an initial temperature distribution given by $u_0(x)$. It reads as follows:

$$\begin{aligned} \frac{\partial}{\partial t}u - \Delta u &= g \qquad \textit{in } \Omega \times J, \\ u &= 0 \qquad \textit{on } \partial\Omega \times J, \\ u|_{t=0} &= u_0 \quad\ \textit{in } \Omega. \end{aligned}$$

Indeed, by setting

$$\begin{aligned} X &= H_0^1(\Omega), & H &= L^2(\Omega), \\ a(u,v) &= \int_\Omega \nabla u \nabla v \; dxdy, & f(v) &= \int_\Omega gv \; dxdy, \end{aligned}$$

this equation can be cast in our setting. Other examples can be found in [30,31].

2.2 DGFEM Time Discretization

We consider the time discretization of problem (1) by the DGFEM, and show that this technique results in an unconditionally stable, implicit single step scheme where arbitrary variations in the time steps and the approximation orders are allowed. We start by introducing time meshes.

A time mesh consists of a partition $\mathcal{M}$ of the time interval $J = (0, T)$ into M time steps $\{I_m\}_{m=1}^M$ given by $I_m = (t_{m-1}, t_m)$ with nodes $0 =: t_0 < t_1 < \ldots < t_{M-1} < t_M := T$. The length of time step I_m is $k_m := t_m - t_{m-1}$, $1 \leq m \leq M$. Furthermore, we introduce a vector $\underline{r}$ which assigns a temporal approximation order r_m to each time step I_m. The discretization parameters $(\mathcal{M}, \underline{r})$ are illustrated in Figure 1.

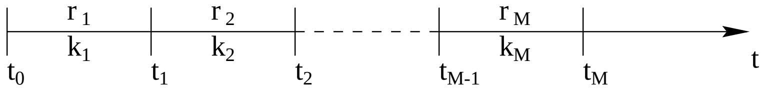

Fig. 1. Partition of J into a time mesh $\mathcal{M}$. The temporal approximation order on time step $I_m = (t_{m-1}, t_m)$ is r_m.

The idea of the DGFEM is to approximate the exact solution u by a (semi)discrete function U which, on time step I_m, consists of a polynomial in t of order r_m and with coefficients in X. These polynomials on the different time steps are not required to be continuous across the time nodes. This allows us to write the DGFEM as a time stepping scheme, see (3). In order to deal with the discontinuities across node t_m we define the left and right handed limits of a function $u : J \to H$ (or $u : J \to X$) to be

$$\begin{aligned} u_m^+ &= \lim_{s\to 0, s>0} u(t_m + s), \qquad 0 \le m \le M-1, \\ u_m^- &= \lim_{s\to 0, s>0} u(t_m - s), \qquad 1 \le m \le M. \end{aligned}$$

Furthermore, the jump of u across t_m is defined as

$$[u]_m := u_m^+ - u_m^-, \qquad 1 \le m \le M.$$

For the exact solution $u \in L^2(J; X) \cap H^1(J; X^*)$ these limits exist in H.

On the time mesh $\mathcal{M}$ we now introduce the space

$$C_b(\mathcal{M}; X) := \{u : J \to X \text{ with } u|_{I_m} \in C_b(I_m; X)\}$$

consisting of bounded continuous functions on each time step. The discontinuous behavior of a function $u \in C_b(\mathcal{M}; X)$ is illustrated in Figure 2; the restriction of u to I_m is denoted by u_m.

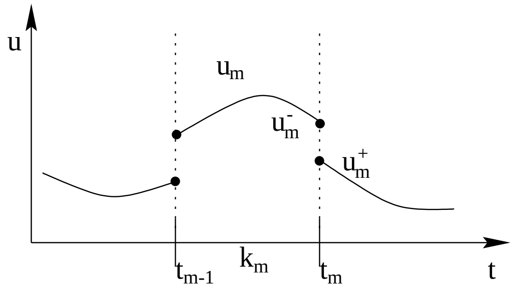

Fig. 2. Discontinuous function $u \in C_b(\mathcal{M}; X)$. The one sided limits u_m^+, u_m^- and the jump $[u]_m$ across the node t_m are sketched symbolically.

We define the bilinear form B_{DG} and the linear form F_{DG} by

$$\begin{aligned} B_{DG}(u, v) &:= \sum_{m=1}^{M} \int_{I_m} \{(u', v)_{X^*\times X} + a(u, v)\} dt \\ &\quad + \sum_{m=2}^{M} ([u]_{m-1}, v_{m-1}^+)_H + (u_0^+, v_0^+)_H, \\ F_{DG}(v) &:= \sum_{m=1}^{M} \int_{I_m} (g(t), v)_{X^*\times X} dt + (u_0, v_0^+)_H. \end{aligned}$$

Note that $\int_{I_m} (u', v)_{X^*\times X} dt = \int_{I_m} (u', v)_H dt$ if $u', v \in L^2(J; H)$. It is straightforward to see by integration by parts that the following lemma holds:

Lemma 2 *Let $u \in L^2(J; X) \cap H^1(J; X^*)$ be the (weak) solution of (1). Then it satisfies $B_{DG}(u, v) = F_{DG}(v)$ for all $v \in C_b(\mathcal{M}; X)$.*

In the DGFEM we now seek the (semi)discrete solution U in the linear subspace $\mathcal{V}^{\underline{r}}(\mathcal{M};X) \subset C_b(\mathcal{M};X)$ which consists of piecewise polynomials in time with coefficients in X

$$\mathcal{V}^{\underline{r}}(\mathcal{M};X) := \{u : J \to X : u|_{I_m} \in \mathcal{P}^{r_m}(I_m;X) \text{ for } 1 \le m \le M\}.$$

If the approximation order r is the same on all time intervals, i.e., $r_m = r$ for $m = 1, \ldots, M$, we simply write $\mathcal{V}^r(\mathcal{M};X)$. Note that the total number of temporal degrees of freedom (dof) is given by $N = \mathrm{NRDOF}(\mathcal{V}^{\underline{r}}(\mathcal{M};X)) := \sum_{m=1}^{M} r_m + 1$ and can be considered as a crude measure for the computational cost of the DGFEM.

For a time discretization $(\mathcal{M}, \underline{r})$ of the interval $J = (0,T)$, the DGFEM for (1) reads now as follows:

$$\text{Find } U \in \mathcal{V}^{\underline{r}}(\mathcal{M};X) \text{ such that } B_{DG}(U,V) = F_{DG}(V) \quad \forall V \in \mathcal{V}^{\underline{r}}(\mathcal{M};X). \tag{2}$$

We cite the following result from [18,35]:

Proposition 3 *The DGFEM (2) has a unique solution $U \in \mathcal{V}^{\underline{r}}(\mathcal{M};X)$. If u is the exact solution of (1), we have the Galerkin orthogonality property, i.e., $B_{DG}(u - U, V) = 0$ for all $V \in \mathcal{V}^{\underline{r}}(\mathcal{M};X)$.*

Due to the discontinuous nature of the test and trial spaces $\mathcal{V}^{\underline{r}}(\mathcal{M};X)$, the DGFEM (2) can be interpreted as an implicit time stepping scheme: The discrete solution U can be found by solving successively for $m = 1, \ldots, M$ the following problems: Find $U_m = U|_{I_m} \in \mathcal{P}^{r_m}(I_m;X)$ such that

$$\begin{aligned}\int_{I_m} \{(U_m', V_m)_H + a(U_m, V_m)\} dt &+ (U_{m-1}^+, V_{m-1}^+)_H \\ = \int_{I_m} (g, V_m)_{X^* \times X}\, dt &+ (U_{m-1}^-, V_{m-1}^+)_H.\end{aligned} \tag{3}$$

for all $V_m \in \mathcal{P}^{r_m}(I_m;X)$. Here, U_{m-1}^- corresponds to the initial datum on time step I_m (we set $U_0^- = u_0$). Note that the initial condition is only satisfied in a weak sense since $[U]_{m-1} \neq 0$ in general.

3 A Priori Error Estimates for the DGFEM Time Discretization

In this section we summarize the a priori error estimates for the DG time stepping method derived in [30,31]. In particular, we show how to resolve temporal singularities in the *hp*-DGFEM. We also mention [32] for a related error analysis of the DGFEM time discretization applied to initial value ODEs.

3.1 hp Approximation Results

We start by collecting a priori estimates for smooth exact solutions.

Theorem 4 *Let the exact solution u of (1) be in $H^{s+1}(J;X)$ for $s \geq 0$, let $r_m = r$ on each time interval I_m, and set $k = \max\{k_m\}$. For the DGFEM approximation $U \in \mathcal{V}^r(\mathcal{M};X)$ we have the error bound*

$$\|u - U\|_{L^2(J;X)} \leq C k^{\min(r,s)+1} r^{-(s+1)} \|u\|_{H^{s+1}(J;X)},$$

with a constant C depending on s.

Let us discuss in more details the implications of Theorem 4. In terms of the total number of time degrees of freedom, $N = \mathrm{NRDOF}(\mathcal{V}^{\underline{r}}(\mathcal{M};X))$, we obtain the following convergence rates:

- For the h-version of the DGFEM where convergence is achieved by decreasing the size k of the time steps at a fixed approximation order r, such that $N \sim \frac{1}{k}$, we have
$$\|u - U\|_{L^2(J;X)} \leq C N^{-\min(r,s)-1}. \tag{4}$$
- For the p-version where convergence is obtained by increasing the approximation order r on a fixed time partition $\mathcal{M}$, such that $N \sim r$, we get
$$\|u - U\|_{L^2(J;X)} \leq C N^{-s-1}. \tag{5}$$

By inspection of the two latter estimates, we can see that the application of the p-version is particularly advantageous if the exact solution is smooth, i.e., if s is larger than r.

In the case where the exact solution is analytic in the closure $\overline{J}$ of J, it can even be shown that the p-version of the DGFEM results in exponential convergence.

Theorem 5 *Let the exact solution u of (1) be analytic in $\overline{J}$. Let $r_m = r$ and let U be the DGFEM solution in $\mathcal{V}^r(\mathcal{M};X)$ on a fixed partition $\mathcal{M}$. Then there holds*

$$\|u - U\|_{L^2(J;X)} \leq C \exp(-br)$$

with constants $C, b > 0$ which are independent of r.

3.2 Start-Up Singularities

However, in practice the regularity assumptions in Theorem 4 and Theorem 5 are unrealistic. This is due to time singularities which may be induced through *non-smooth initial data* or *discontinuities in the right hand side g.*

To analyze the structure of start-up singularities at $t = 0$, we describe the regularity of the initial datum u_0 in terms of intermediate spaces $H_\theta = (H, X)_{\theta,2}$, $0 \leq \theta \leq 1$, defined by the K-method of interpolation [22]. The parameter θ measures the compatibility of u_0 with respect to a continuous scale of intermediate spaces $X \subseteq H_\theta \subseteq H$. In the limiting case $\theta = 0$, we have $u_0 \in H$, corresponding to completely incompatible data, whereas for $\theta = 1$ we have $u_0 \in X$, describing the compatible case. By the use of Fourier series techniques (if L is selfadjoint) or of classical semigroup theory (if L is non-selfadjoint) the subsequent analyticity properties of the exact solution are obtained:

Theorem 6 *Let the right hand side g in (1) be piecewise analytic (in time) and let $u_0 \in H_\theta$ for some $0 \leq \theta \leq 1$. Then the solution u of (1) satisfies*

$$\|u^{(l)}(t)\|_X^2 \leq C d^{2l} \Gamma(2l + 2) t^{-(2l+1)+\theta}$$

in the vicinity of $t = 0$.

Theorem 6 gives a precise characterization of the start-up singularity induced by incompatible initial data. Results of this type have been known, see, e.g, [35]. However, the (basically sharp) explicit dependence of the estimates on l, t and θ is crucial in the *hp*-version context.

3.2.1 Exponential Convergence

We address the resolution of time singularities as in Theorem 6 by the *hp*-version DGFEM. For simplicity, we confine ourselves to the time interval $J = (0, 1)$ and start by introducing geometric time partitions.

Definition 7 *The (basic) geometric partition $\mathcal{M}_{n,\sigma} = \{I_m\}_{m=1}^{n+1}$ of $J = (0, 1)$ with grading factor $\sigma \in (0, 1)$ and $n+1$ time intervals I_m is given by the nodes*

$$t_0 = 0, \qquad t_m = \sigma^{n-m+1}, \qquad 1 \leq m \leq n + 1.$$

On geometric time partitions $\mathcal{M}_{n,\sigma}$ we consider time approximation order distributions $\underline{r}$ which increase from layer to layer:

Definition 8 *A polynomial degree vector $\underline{r} = \{r_m\}_{m=1}^M$ is called linear with slope $\mu > 0$ on the geometric partition $\mathcal{M}_{n,\sigma}$ if $r_m = \lfloor \mu m \rfloor$ for $1 \leq m \leq n+1$.*

Figure 3 gives an example of a linearly increasing degree vector $\underline{r}$. The combination of a geometrically refined partition $\mathcal{M}_{n,\sigma}$ with a linear approximation order vector $\underline{r}$ allows us to recover exponential rates of convergence, even when temporal singularities are present as in Theorem 6. The following theorem gives a precise formulation of this fact.

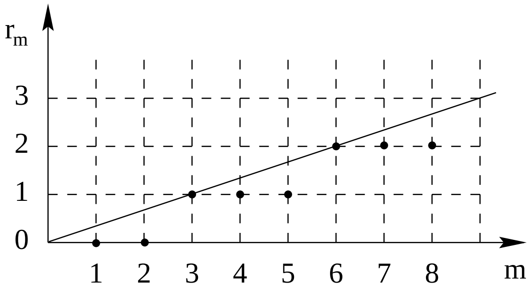

Fig. 3. Example of a linear polynomial degree vector with slope $\mu = 1/3$ on 8 time steps. The resulting degree vector is $\underline{r} = (0, 0, 1, 1, 1, 2, 2, 2)$.

Theorem 9 *Let u be the exact solution of (1) with $J = (0, 1)$, initial datum $u_0 \in H_\theta$ for some $0 < \theta \leq 1$ and piecewise analytic right hand side g. Let U be the DGFEM solution in $\mathcal{V}^{\underline{r}}(\mathcal{M}_{n,\sigma}; X)$ for a geometric partition $\mathcal{M}_{n,\sigma}$ and a linearly increasing degree vector $\underline{r}$. Then there exists $\mu_0 > 0$ such that for all linear polynomial degree vectors $\underline{r} = \{r_m\}_{m=1}^{n+1}$ with slope $\mu \geq \mu_0$ we have the error estimate*

$$\|u - U\|_{L^2(J;X)} \leq C \exp(-bN^{\frac{1}{2}})$$

with constants C and b independent of $N = \mathrm{NRDOF}(\mathcal{V}^{\underline{r}}(\mathcal{M}_{n,\sigma}; X))$.

3.2.2 Algebraically Graded Time Steps

In the h-version DGFEM on quasiuniform partitions the best possible convergence rate of N^{-r-1} is lost if the solution u is not smooth enough in time, i.e., if $s < r$ in (4). In the h-FEM graded meshes are used to compensate for this loss of convergence. The same mechanism works for the DGFEM time stepping as well. To show this, we assume again that $J = (0, 1)$.

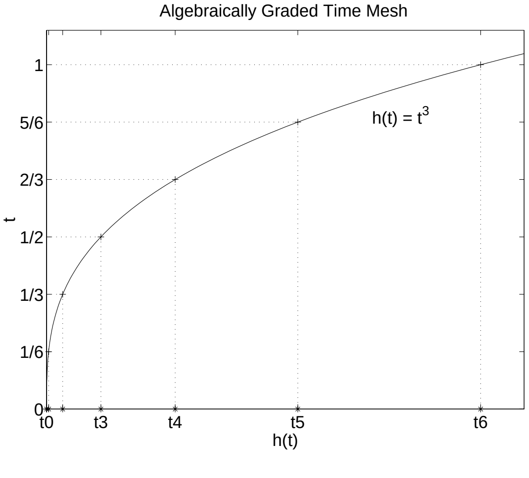

Fig. 4. Example of an algebraically graded time mesh $\mathcal{M}$ for 6 time steps in the interval $J = (0, 1)$ with grading function $h(t) = t^3$.

Definition 10 *An algebraically graded temporal mesh $\mathcal{M}$ is defined by a grading function $h : [0,1] \to [0,1]$ which is strictly increasing and satisfies*

$$h \in C^0([0,1]) \cap C^1((0,1)), \qquad h(0) = 0, \qquad h(1) = 1.$$

The nodes in $\mathcal{M}$ are given by $t_m = h(\frac{m}{M})$, $m = 0, \ldots, M(\mathcal{M})$.

Figure 4 shows an example of an algebraically graded mesh. Note that for time meshes with uniform approximation order r the number of time steps M and the number of temporal dof N are proportional, in fact $N = (r+1)M$. The following theorem states that we can recover the optimal h-version convergence rate by the use of an appropriate grading function h.

Theorem 11 *Let u be the exact solution of (1) with $J = (0,1)$, initial datum $u_0 \in H_\theta$ for some $0 < \theta \leq 1$ and piecewise analytic right hand side g. Consider the h-version DGFEM method at a fixed approximation order r on the graded time mesh $\mathcal{M}$ with M time steps given by the grading function $h(t) = t^{\frac{2r+3}{\theta}}$. Let $N = \mathrm{NRDOF}(\mathcal{V}^r(\mathcal{M}; X))$. Then, as $M \to \infty$ or $N \to \infty$, there holds for the DGFEM solution $U \in \mathcal{V}^r(\mathcal{M}; X)$*

$$\|u - U\|_{L^2(J;X)} \leq CM^{-(r+1)} \qquad \text{or} \qquad \|u - U\|_{L^2(J;X)} \leq CN^{-(r+1)}$$

with C depending only on u_0, g and r.

4 A Fully Discrete *hp* Scheme

In practice, the elliptic spatial operator L in (1) has to be discretized as well, in order to obtain a fully discrete scheme for parabolic problems. In this section we introduce an *hp*-FEM for this spatial discretization, based on a finite element space $X_D \subset X$ of dimension $\dim(X_D) = D < \infty$. To this end, we choose a basis $\{\varphi_{i,m}(t)\}_{i=0}^{r_m}$ of $\mathcal{P}^{r_m}(I_m)$ and write the semidiscrete solution U on time step I_m as

$$U_m = U|_{I_m} = \sum_{j=0}^{r_m} u_{j,m} \varphi_{j,m},$$

with coefficients $\{u_{j,m}\}$ in the continuous space X. We then show in Section 4.1 that the DGFEM actually amounts to solving a system of $r_m + 1$ coupled elliptic reaction diffusion equations for the coefficients $u_{j,m}$. To obtain a fully discrete solution these equations are now discretized in the FE space X_D. In Section 4.2 we present a straightforward way to approximate the coefficients $\{u_{j,m}\} \subset X$ by discrete FE coefficients $\{u_{j,m}^{FE}\} \subset X_D$. On time step I_m the fully discrete solution is then of the form

$$U_m^{FE} = \sum_{j=0}^{r_m} u_{j,m}^{FE} \varphi_{j,m}.$$

It turns out, however, that the straightforward approach may be very costly in terms of computational time; thus, in Section 4.3 and Section 4.4 we introduce and discuss a decoupling procedure that substantially decreases the computational cost of the discretization of the spatial problems. Section 4.5 and Section 4.6 address local static condensation and present the complete algorithmic pattern of our fully discrete *hp* method.

4.1 Time Shape Functions and Spatial Problems

We focus on the solution of problem (3) on time step $I_m = (t_{m-1}, t_m)$ with approximation order r_m. The initial condition U_{m-1}^- as well as the right hand side $g(t)$ are assumed to be given.

Let $\{\widehat{\varphi}_i\}_{i=0}^r$ be a basis of the polynomial space $\mathcal{P}^r((-1,1))$. We also refer to $\{\widehat{\varphi}_i\}_{i=0}^r$ as reference time shape functions. On the interval (t_{m-1}, t_m), the time shape functions $\varphi_{i,m}(t)$ are then defined as $\varphi_{i,m} \circ F_m(\hat{t}) = \widehat{\varphi}_i(\hat{t})$, where F_m is the mapping from the reference interval $(-1,1)$ to (t_{m-1}, t_m) given by

$$t = F_m(\hat{t}) := \frac{1}{2}(t_{m-1} + t_m) + \frac{1}{2}\hat{t}k_m. \tag{6}$$

Since the semidiscrete DGFEM approximation U_m and the test function V_m in (3) are both in the polynomial space $\mathcal{P}^{r_m}(I_m; X)$, they can uniquely be written in the basis $\{\varphi_{i,m}\}_{i=0}^{r_m}$ as

$$U_m = \sum_{j=0}^{r_m} u_{j,m}\varphi_{j,m}, \qquad\qquad V_m = \sum_{i=0}^{r_m} v_{i,m}\varphi_{i,m}, \tag{7}$$

with coefficients $\{u_{j,m}\}$, $\{v_{i,m}\}$ in X. In the following, we choose normalized Legendre polynomials as reference time shape functions, that is

$$\widehat{\varphi}_j(\hat{t}) = \sqrt{(j + 1/2)} \cdot L_j(\hat{t}), \quad j \geq 0, \tag{8}$$

with L_j being the usual Legendre polynomial of degree j on $(-1,1)$.

Example 12 *The first five reference time shape functions of the form (8) are*

$$\begin{aligned}
\hat{\varphi}_0(\hat{t}) &= \sqrt{1/2},\\
\hat{\varphi}_1(\hat{t}) &= \sqrt{3/2}\cdot\hat{t},\\
\hat{\varphi}_2(\hat{t}) &= \sqrt{5/2}\cdot(3\hat{t}^2-1)/2,\\
\hat{\varphi}_3(\hat{t}) &= \sqrt{7/2}\cdot(5\hat{t}^3-3\hat{t})/2,\\
\hat{\varphi}_4(\hat{t}) &= \sqrt{9/2}\cdot(35\hat{t}^4-30\hat{t}^2+3)/8.
\end{aligned}$$

In order to avoid cumbersome notation, we consider for the rest of this section a generic time step $I = (\alpha, \beta)$, omit the index m and write $\{\varphi_i\}$ for the time shape functions, r for the approximation order, $\{u_j\}$ and $\{v_i\}$ for the coefficients in (7), F for the element mapping, and U_{init} for the initial datum at $t = \alpha$.

By inserting the ansatz (7) in the time stepping scheme (3), the DGFEM amounts to the following system of equations for $\{u_j\} \subset X$: Find coefficients $\{u_j\}_{j=0}^r \subset X$ such that

$$\begin{aligned}
&\sum_{i,j=0}^{r}\{[\overbrace{\int_I \varphi_j'\varphi_i dt + \varphi_j^+(\alpha)\varphi_i^+(\alpha)}^{A_{ij}}](u_j,v_i)_H + [\overbrace{\int_I \varphi_j\varphi_i dt}^{B_{ij}}]\,a(u_j,v_i)\}\\
&= \sum_{i=0}^{r}\{\overbrace{(\int_I g\varphi_i dt, v_i)_H}^{f_i^1} + \overbrace{(U_{init}, v_i)_H\varphi_i^+(\alpha)}^{f_i^2}\} \qquad \text{for all} \quad \{v_i\}_{i=0}^r \subset X.
\end{aligned} \tag{9}$$

Here, $\varphi_i^+(\alpha) = \lim_{t\to\alpha, t>\alpha}\varphi_i(t)$. Introducing the following abbreviations (in terms of the time reference shape functions $\{\widehat{\varphi}_i(\hat{t})\}$)

$$\begin{aligned}
\hat{A}_{ij} &:= \int_{-1}^{1}\hat{\varphi}_j'\hat{\varphi}_i d\hat{t} + \hat{\varphi}_j^+(-1)\hat{\varphi}_i^+(-1),\\
\hat{B}_{ij} &:= \int_{-1}^{1}\hat{\varphi}_j\hat{\varphi}_i d\hat{t},\\
\hat{f}_i^1(v) &:= (\hat{l}_i^1, v)_H, \qquad & \hat{l}_i^1 &:= \int_{-1}^{1}(g\circ F)\widehat{\varphi}_i d\hat{t},\\
\hat{f}_i^2(v) &:= (\hat{l}_i^2, v)_H, \qquad & \hat{l}_i^2 &:= U_{init}\widehat{\varphi}_i(-1),
\end{aligned} \tag{10}$$

we can write the system (9) in the compact form:

$$\begin{gathered}
\text{Find coefficients } \{u_j\}_{j=0}^r \subset X \text{ such that for all } \{v_i\}_{i=0}^r \subset X\\
\sum_{i,j=0}^{r}\hat{A}_{ij}(u_j,v_i)_H + \frac{k}{2}\hat{B}_{ij}a(u_j,v_i) = \sum_{i=0}^{r}\frac{k}{2}\hat{f}_i^1(v_i) + \hat{f}_i^2(v_i).
\end{gathered} \tag{11}$$

The action of the right hand side g is represented by the load $\hat{f}_i^1$, whereas the second load $\hat{f}_i^2$ accounts for the initial datum U_{init}. Note that on the first time step we have $U_{init} = u_0$.

Remark 13 The matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ in (10) are independent of the time step I and can be calculated in a preprocessing step. Their size, however, depends on the approximation order r on time step I.

The ideal choice of time shape functions $\hat{\varphi}_i$ would be the one where $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ diagonalize simultaneously. The system (11) would in this case decouple into $r+1$ independent scalar equations. In Section 4.3, we introduce such a decoupling procedure which, however, requires to switch over to complex arithmetic.

Example 14 *For the time shape functions in (8) the matrix* $\hat{\mathbf{B}}$ *is the identity matrix and* $\hat{\mathbf{A}}$ *is, for* $0 \le i, j \le r = 5$*, given by the matrix*

$$\hat{\mathbf{A}} = \begin{pmatrix} 0.5000 & 0.8660 & 1.1180 & 1.3228 & 1.5000 & 1.6583 \\ -0.8660 & 1.5000 & 1.9365 & 2.2913 & 2.5981 & 2.8723 \\ 1.1180 & -1.9365 & 2.5000 & 2.9580 & 3.3541 & 3.7081 \\ -1.3229 & 2.2913 & -2.9580 & 3.5000 & 3.9686 & 4.3875 \\ 1.5000 & -2.5981 & 3.3541 & -3.9686 & 4.5000 & 4.9749 \\ -1.6583 & 2.8723 & -3.7081 & 4.3875 & -4.9749 & 5.5000 \end{pmatrix}. \qquad (12)$$

The matrix $\hat{\mathbf{A}}$ *is hierarchical in* r*, i.e., to obtain* $\hat{\mathbf{A}}$ *for* $r \le 5$*, we simply take the submatrix of (12) containing the first* $r+1$ *rows and columns, e.g., for* $r = 2$ *we have*

$$\hat{\mathbf{A}} = \begin{pmatrix} 0.5 & 0.8660 & 1.1180 \\ -0.8660 & 1.5 & 1.9365 \\ 1.1180 & -1.9365 & 2.5 \end{pmatrix}.$$

Note that with the shape functions in (8) the strong form of the system (11) reads

$$\sum_{j=0}^{r} \hat{A}_{ij} u_j + \frac{k}{2} \delta_{ij} L u_j = \frac{k}{2} \hat{l}_i^1 + \hat{l}_i^2, \qquad i = 0, \dots, r, \qquad (13)$$

or, in matrix notation,

$$\hat{\mathbf{A}} \vec{u} + \frac{k}{2} [\delta_{ij}] L \vec{u} = \frac{k}{2} \vec{l^1} + \vec{l^2}, \qquad (14)$$

with $\vec{u} = (u_0, \dots, u_r)^T$, $\vec{l^i} = (\hat{l}_0^i, \dots, \hat{l}_r^i)^T$ for $i = 1, 2$.

Remark 15 For the time approximation order $r = 0$, the spatial problems in (13) are identical to the ones that are obtained with the well known implicit Euler time stepping scheme. For higher time approximation orders $r > 0$, the DGFEM is equivalent to certain implicit Runge-Kutta schemes [21].

4.2 Direct Spatial Discretization

We use standard *hp*-FEM techniques to discretize problem (11) in space (for details about *hp*-FEM theory, we refer to [33] and the references therein). We choose a finite dimensional subspace $X_D \subset X$ of dimension D and look for finite element approximations $\{u_j^{FE}\} \subset X_D$ to the continuous coefficients $\{u_j\} \subset X$. To this end, let $\{s_j\}_{j=1}^D$ be a basis of the finite element space X_D, then we write the trial and test functions u_j^{FE}, v_j^{FE}, $j = 0, \ldots, r+1$, as linear combinations of basis functions s_l with unknown coefficients $u_j^l, v_j^l \in \mathbb{R}$, i.e.,

$$u_j^{FE} = \sum_{l=1}^{D} u_j^l s_l(x), \qquad v_i^{FE} = \sum_{k=1}^{D} v_i^k s_k(x). \tag{15}$$

We insert this ansatz into (11) and define the mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{S}$ to be

$$\mathbf{M} := \{(s_l, s_k)_H\}_{l,k=1}^D, \qquad \mathbf{S} := \{a(s_l, s_k)\}_{l,k=1}^D. \tag{16}$$

The fully discrete system that we obtain for the unknown coefficient vectors $\vec{u}_j = (u_j^1, u_j^2, \ldots, u_j^D)^T \in \mathbb{R}^D$ has the generic structure

$$\begin{bmatrix} \hat{A}_{00}\mathbf{M} + \frac{k}{2}\mathbf{S} & \cdots & \hat{A}_{0r}\mathbf{M} \\ \vdots & \ddots & \vdots \\ \hat{A}_{r0}\mathbf{M} & \cdots & \hat{A}_{rr}\mathbf{M} + \frac{k}{2}\mathbf{S} \end{bmatrix} \begin{bmatrix} \vec{u}_0 \\ \vdots \\ \vec{u}_r \end{bmatrix} = \frac{k}{2} \begin{bmatrix} \vec{f}_0^1 \\ \vdots \\ \vec{f}_r^1 \end{bmatrix} + \begin{bmatrix} \vec{f}_0^2 \\ \vdots \\ \vec{f}_r^2 \end{bmatrix} \tag{17}$$

with load vectors

$$\begin{aligned} \vec{f}_j^1 &= (\hat{f}_j^1(s_1), \hat{f}_j^1(s_2), \ldots, \hat{f}_j^1(s_D))^T, \\ \vec{f}_j^2 &= (\hat{f}_j^2(s_1), \hat{f}_j^2(s_2), \ldots, \hat{f}_j^2(s_D))^T. \end{aligned}$$

Note that a linear system of the above type (17) with dimension $(r+1)D$ has to be solved on every time step. The use of efficient sparse linear system solvers is therefore mandatory. We emphasize that the work to set up the global matrix is reduced due to the repeated appearance of $\mathbf{M}$ and $\mathbf{S}$.

4.3 Decoupling

In terms of computing time it is very costly to solve a fully discrete system of the form (17). It would therefore be desirable to decouple the system (14) into $r+1$ scalar problems that could be solved independently. Unfortunately, this seems not to be possible with time shape functions in $\mathbb{R}$. However, numerical experiments show that the matrix $\hat{\mathbf{A}}$ in (10), evaluated for the Legendre time shape functions, is diagonalizable in $\mathbb{C}$ at least for $0 \leq r \leq 100$: There exists a matrix $\mathbf{Q} \in \mathbb{C}^{(r+1)\times(r+1)}$ such that

$$\mathbf{Q}^{-1}\hat{\mathbf{A}}\mathbf{Q} = \hat{\mathbf{T}} = \mathrm{diag}(\lambda_1^{(r)}, ..., \lambda_{r+1}^{(r)}), \tag{18}$$

with pairwise complex conjugate eigenvalues $\lambda_j = \lambda_j^{(r)}$. These matrices allow us to decouple the system (14) in the following way

$$\underbrace{\mathbf{Q}^{-1}\hat{\mathbf{A}}\mathbf{Q}}_{\hat{\mathbf{T}}}\underbrace{\mathbf{Q}^{-1}\vec{u}}_{\vec{w}}+\frac{k}{2}L\underbrace{\mathbf{Q}^{-1}\vec{u}}_{\vec{w}} = \mathbf{Q}^{-1}(\frac{k}{2}\vec{l}^1 + \vec{l}^2), \tag{19}$$

and we obtain, with $\vec{w} = \mathbf{Q}^{-1}\vec{u}$,

$$\hat{\mathbf{T}}\vec{w} + \frac{k}{2}L\vec{w} = \mathbf{Q}^{-1}(\frac{k}{2}\vec{l}^1 + \vec{l}^2). \tag{20}$$

Equivalently, we can write the $r+1$ decoupled equations as

$$\lambda_j w_j + \frac{k}{2}Lw_j = [\frac{k}{2}\mathbf{Q}^{-1}\vec{l}^1 + \mathbf{Q}^{-1}\vec{l}^2]_j \quad \text{for } 0 \leq j \leq r. \tag{21}$$

In practical applications we limit ourselves to time approximation orders of $r \leq 12 = r_{max}$. The corresponding matrices $\mathbf{Q}$, $\mathbf{Q}^{-1}$ and $\hat{\mathbf{T}}$, for $0 \leq r \leq r_{max}$, can be computed and stored in a preprocessing step since they do not vary during the time stepping.

4.4 Spatial Discretization after Decoupling

The decoupling process requires the solution of the $r+1$ independent equations (21) in $\mathbb{C}$. In this section we consider the spatial discretization of these equations by *hp*-FEM techniques. The standard weak formulation of (21) is

$$\begin{aligned} &\text{Find } w_j \in X \text{ such that for all } v \in X\text{:} \\ &b_j(w_j, v) := \lambda_j(w_j, v)_H + \frac{k}{2}a(w_j, v) = \hat{f}_j^c(v), \end{aligned}$$

where the composed, transformed load $\hat{f}^c_j$ is given by

$$\hat{f}^c_j = [\frac{k}{2}Q^{-1}\vec{f}^1 + Q^{-1}\vec{f}^2]_j \qquad \text{with} \qquad \vec{f}^i = (\hat{f}^i_0, \hat{f}^i_1, ..., \hat{f}^i_r)^T, \quad i = 1, 2.$$

The corresponding FEM approximation is

$$\text{Find } w^{FE}_j \in X_D \quad \text{such that} \quad b_j(w^{FE}_j, v) = \hat{f}^c_j(v) \quad \text{for all} \quad v \in X_D. \tag{22}$$

Inserting in (22) an ansatz of the type (15) for w^{FE}_j yields the following linear system for the unknown coefficient vector $\vec{w}_j = (w^1_j, w^2_j, .., w^D_j)^T \in \mathbb{C}^D$

$$\overbrace{[\lambda_j \mathbf{M} + \frac{k}{2}\mathbf{S}]}^{\mathbf{G}_j}\, \vec{w}_j = \vec{f^c_j} \tag{23}$$

with $\vec{f^c_j} = (\hat{f}^c_j(s_1), \hat{f}^c_j(s_2), ..., \hat{f}^c_j(s_D))^T$. We call the matrix $\mathbf{G}_j := \lambda_j \mathbf{M} + \frac{k}{2}\mathbf{S}$ the global matrix of the specific jth spatial system in (21) with the mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{S}$ defined as in (16). We get the coefficients $\vec{u}_j$ for the functions u^{FE}_j by applying the backtransformation

$$\vec{u}_j = \sum_{i=1}^{r+1} Q_{ji}\vec{w}_j. \tag{24}$$

We give the following example to clarify the structure of the load vector $\vec{f^c_j}$ in (23).

Example 16 *Let the time approximation order be $r = 1$, then the load vector $\vec{f^c_j}$ for the system $j = 1$ takes the form*

$$\begin{bmatrix} \hat{f}^c_{1,1} \\ \vdots \\ \hat{f}^c_{1,D} \end{bmatrix} = [Q^{-1}]_{11} \begin{bmatrix} \frac{k}{2}\hat{f}^1_1(s_1) + \hat{f}^2_1(s_1) \\ \vdots \\ \frac{k}{2}\hat{f}^1_1(s_D) + \hat{f}^2_1(s_D) \end{bmatrix} + [Q^{-1}]_{12} \begin{bmatrix} \frac{k}{2}\hat{f}^1_2(s_1) + \hat{f}^2_2(s_1) \\ \vdots \\ \frac{k}{2}\hat{f}^1_2(s_D) + \hat{f}^2_2(s_D) \end{bmatrix}.$$

We emphasize that the load vector for system $j = 1$ involves also terms containing $\hat{f}^i_2$. This is important with regard to parallelization.

We also remark that each of the decoupled equations in (21) corresponds to a singularly perturbed problem of the form

$$\varepsilon^2 Lw + w = f \tag{25}$$

where $\varepsilon = \varepsilon^{(r)}_{j,m} = \sqrt{k_m/(2\lambda_j)} \in \mathbb{C}$, with $\lambda_j = \lambda^{(r)}_j$ of (18), $\sqrt{\cdot}$ being the usual principal branch of the square root taken to be positive on $(0, \infty)$.

The following lemma, proved in [31], analyzes the dependence of λ_j (and therefore also ε) on r; we see that the modulus $|\varepsilon|$ can approach zero as $r \to \infty$.

Lemma 17 *Let $\lambda_j \in \mathbb{C}$ be an eigenvalue of the matrix $\hat{A}$ in (10). Then*

$$\operatorname{Re} \lambda_j \leq C_1 \max(1, r^2) \quad \textit{and} \quad 0 < C_2 \leq |\lambda_j| \leq C_3 \max(1, r^2)$$

with constants independent of $r \in \mathbb{N}_0$.

The small parameter ε in (25) can cause difficulties due to the appearance of boundary layers in the exact solution. In [24,25], however, it is shown that the *hp*-version FEM for problems of the form (25) leads to robust exponential convergence rates (independent of the perturbation parameter ε) provided that certain mesh design principles are followed.

4.5 Local Static Condensation

One of the major motivations for using the DGFEM time stepping method is its high accuracy. Since it is not reasonable to have high accuracy only in the time discretization, it is natural to deal with approximation orders $p \geq 2$ in the spatial discretization. This implies that the global mass and stiffness matrix $\mathbf{M}$ and $\mathbf{S}$ can become large, even if only a small number of spatial elements is used. In the case of quadrilateral elements with uniform approximation order p the number of external dof per element grows like $d_e = 4p$, while the number of internal dof grows like $d_i = (p-1)^2$. It is therefore advantageous to eliminate the internal dof by local static condensation. To do so, the dof in all element matrices $\mathbf{E}^{[k]} = \alpha \mathbf{M}^{[k]} + \beta \mathbf{S}^{[k]}$, k denoting the element number, are sorted in the following way

$$\begin{bmatrix} \mathbf{E}_{ee} & \mathbf{E}_{ei} \\ \mathbf{E}_{ie} & \mathbf{E}_{ii} \end{bmatrix} \begin{bmatrix} \vec{x}_e \\ \vec{x}_i \end{bmatrix} = \begin{bmatrix} \vec{f}_e \\ \vec{f}_i \end{bmatrix},$$

where the subscript e stands for external and i for internal. It is then straightforward to see that we can solve the condensed system $\mathbf{E}_c x_e = \vec{f}_c$ instead of the full system, with the condensed element matrix $\mathbf{E}_c$ and the condensed right hand side $\vec{f}_c$ being

$$\mathbf{E}_c = \mathbf{E}_{ee} - \mathbf{E}_{ei} \overbrace{\mathbf{E}_{ii}^{-1} \mathbf{E}_{ie}}^{\mathbf{C}}, \qquad \vec{f}_c = \vec{f}_e - \mathbf{E}_{ei} \overbrace{\mathbf{E}_{ii}^{-1} \vec{f}_i}^{\vec{c}}.$$

The condensed matrices $\mathbf{E}_c$ and the loads $\vec{f}_c$ are assembled to a global system for the external dof $\vec{x}_e$. After solving this system, one obtains the internal dof $\vec{x}_i$ by a backsolve:

$$\vec{x}_i = \mathbf{E}_{ii}^{-1} \vec{f}_i - \mathbf{E}_{ii}^{-1} \mathbf{E}_{ie} \vec{x}_e. \tag{26}$$

We can avoid to compute $\mathbf{E}_{ii}^{-1}$ explicitly by solving $\mathbf{E}_{ii}\vec{y} = [\mathbf{E}_{ie}\vdots\vec{f}_i]$ for $\vec{y}$ instead. If enough memory is available, the results can be stored (for each element) in the matrix $\mathbf{C}$ and in the vector $\vec{c}$. The backsolve (26) is then reduced to a simple matrix-vector multiplication and a vector addition, that is $\vec{x}_i = \vec{c} - \mathbf{C}\vec{x}_e$.

4.6 Implementation

We now describe the fully discrete DGFEM time stepping algorithm in the case where we decouple the spatial systems (13). We compute and store the matrix $\hat{\mathbf{A}}$ and the matrices $\mathbf{Q}$, $\mathbf{Q^{-1}}$, $\hat{\mathbf{T}}$ in a preprocessing step. Since in practical applications one is only interested in approximation orders of, e.g., $r \leq r_{max} = 12$, this does not require much memory. Furthermore, the elemental mass and stiffness matrices are (up to scalars $\alpha, \beta \in \mathbb{C}$) the same for the problems (23) and can therefore be computed in advance as well. This is summarized in Algorithm 18.

Algorithm 18 DGFEM Preprocessing

```
Compute the matrix $\hat{A}$
Compute the matrices $Q, Q^{-1}, \hat{T}$ for all $r = 0..., r_{max}$
Do $i = 1$, #elements in spatial discretization
    Compute the local stiffness matrix, store it in StiffM(i)
    Compute the local mass matrix, store it in MassM(i)
Enddo
```

The actual time stepping using the decoupling strategy is described in Algorithm 19 with the arrays `ElemM(:)`, `MassM(:)`, `StiffM(:)` and `LoadV(:)` of element matrices and load vectors, respectively. Note that if no local static condensation is performed, it is advantageous to assemble the element mass and stiffness matrices separately into $\mathbf{M}$ and $\mathbf{S}$. Building the global matrix $\mathbf{G}_{j,m}$ for one of the systems $j = 0, \ldots, r_m$ consists then only of computing $\mathbf{G}_{j,m} = \lambda_j \mathbf{M} + \frac{k_m}{2}\mathbf{S}$. In the case of uniform time steps and a constant time approximation order $r_m = r$ (h-version DGFEM) the matrices $\mathbf{G}_{j,m}$, $j = 0, \ldots, r$, are the same for all time steps such that we have to compute their LU decomposition only on the first time step. On all subsequent time steps, we just have to perform backsolves with different right hand sides.

Algorithm 19 DGFEM Time Stepping (with decoupling)

```
Do $m = 1, M$ (loop over all time steps m)
    Do $j = 1, r_m + 1$ (loop over spatial systems $j$)
       Do $i = 1$, #elements in space (build global matrix)
          ElemM(i) $= \lambda_j \cdot$ MassM(i) $+ \frac{k}{2} \cdot$ StiffM(i)
          Compute the right hand side, store it in LoadV(i)
          Condense ElemM(i), LoadV(i),
             Store matrix $\mathbf{C}$ and vector $\vec{c}$ for backsolve
          Implement boundary conditions in ElemM(i), LoadV(i)
          Assemble ElemM(i) into the global matrix $\mathbf{G}_{j,m}$
          Assemble LoadV(i) into the global rhs $\vec{f}^c_{j,m}$
       Enddo
       Solve the system $\mathbf{G}_{j,m}\vec{w}_j = \vec{f}^c_{j,m}$ for the global external dof
       Do $i = 1$, #elements
          Do $k = 1, r_m + 1$
          Local backsolve $\vec{w}^i_j|_i = \vec{c} - \mathbf{C}\vec{w}^e_j|_i$.
          Local backtransformation $\vec{u}|_i = \vec{u}|_i + [Q]_{ik}\vec{w}|_i$
          Enddo
       Enddo
    Enddo
Enddo
```

5 Parallelization Strategies

In this section we briefly focus on some parallelization aspects which, of course, are of essential importance for real engineering applications. First of all, the integration of the element matrices as well as their static condensation is perfectly parallelizable, since for this computation no communication is needed. The distribution of the element matrices on the processors can either be done dynamically, i.e., by a client-server model where the work load of the processors is automatically balanced, or by a domain decomposition method (this holds for both, shared and distributed memory architectures). The client-server model is already implemented and tested in the code PHP90 which is an extension of the code HP90 [10] that we applied in this work (see also Section 6).

5.1 Shared Memory Computers

During the time stepping, we mainly have to build and compute the $r + 1$ systems (23) on each time step. Obviously, on shared memory machines, we have an extremely simple and yet effective possibility to parallelize this task: We build the global systems $\mathbf{G}_{j,m}\vec{w}_{j,m} = \vec{l}_{j,m}$ sequentially and take advantage of the capabilities of a parallel solver such as *PARDISO* [28,29] to solve the systems. *PARDISO* is a scalable parallel direct solver, designed to solve sparse symmetric or structurally symmetric linear systems on shared memory multiprocessors. It features state-of-the-art techniques for the reordering and fill-in reduction, utilizes block techniques for Level BLAS-3 use and optimizes the memory and processor locality. In the case of large scale applications, it is conceivable to solve the $r+1$ systems one after the other (but each of them in parallel). Or, if enough processors and memory are available, each of the systems can be solved by a group of processors in parallel.

5.2 Distributed Memory Computers

Let us assume that the number of processors q fulfills $q \gg r+1$ such that we can form $r+1$ processor groups. One specific group will be called master group. Then a possible parallel algorithm has the following frame:

a) Distribute all elements among the processor groups.
b) In all groups j: Integrate the element matrices, send the results to the master group.
c) In the master group: Assemble the mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{S}$.
d) In the master group: Broadcast M, S to all groups (one-to-all send).
e) In all groups j: Compute load $\vec{v}_j = \frac{k}{2}\vec{f}_j^1 + \vec{f}_j^2$.
f) In all groups j: Send load $\vec{v}_j$ to all others, receive loads $\vec{v}_i, i \neq j$ (all-to-all send), build transformed load $\vec{f}_j^c = \sum_{i=1}^{r+1} Q_{ji}^{-1} \vec{v}_j$.
g) In all groups j: Solve system $[\lambda_j \mathbf{M} + \frac{k}{2}\mathbf{S}]\vec{w}_j = \vec{f}_j^c$
h) In all groups j : Send solution $\vec{w}_j$ to master group (all-to-one send).
i) Master group : Compute backtransformed solution $\vec{u}_j = \sum_{i=1}^{r+1} Q_{ji}\vec{w}_i$ and broadcast it to all groups (one-to-all send).

An important aspect of this parallel algorithm is the efficient solution of the linear systems within one processor group, see, e.g., [27].

6 Numerical Results

6.1 The Model Problems

Starting from the code HP90 [10] which is designed to solve general elliptic problems in a *hp*-FEM context, we have developed a new code which is able to solve general parabolic problems using the *hp*-DGFEM method for the time discretization. As a test problem, we chose the standard heat equation from Example 1 with $d = 2$, the computational domain being the unit square $\Omega = (0,1)^2$ and the time interval being $J = (0, 0.1)$. We only integrate to the time $t_M = 0.1$ because the solutions that we consider are strongly smoothened out in time and we are above all interested in the resolution of the time singularities at $t_0 = 0$. We are using three problems with different initial conditions and right hand sides:

- Problem (I): Here, we choose the initial datum $u_0(x, y) = \sin(\pi x)\sin(\pi y)$ and right hand side $g = 0$. u_0 is actually the first eigenfunction of the Laplacian and compatible with being in $H_0^1(\Omega)$. The corresponding exact solution $u(x, y, t)$ is smooth in space and time and given as

$$u(x, y, t) = \exp(-2\pi^2 t)\sin(\pi x)\sin(\pi y).$$

- Problem (II): We take initial datum $u_0(x, y) = x(1-x)y(1-y)$ and $g = 0$. u_0 is also compatible with being in $H_0^1(\Omega)$. The exact solution $u(x, t)$ can be represented as a Fourier series with coefficients $\{a_{kl}\}$:

$$\begin{aligned} u(x,t) &= \sum_{l=1}^{\infty}\sum_{k=1}^{\infty} a_{kl}\exp(-\pi^2(l^2+k^2)t)\sin(l\pi x)\sin(k\pi y), \\ a_{kl} &= 16\frac{(1-\cos(l\pi))(1-\cos(k\pi))}{l^3k^3\pi^6}. \end{aligned}$$

- Problem (III): Here, $u_0(x, y) = 0$ and $g = 2t^\alpha(x(1-x) + y(1-y)) + x(1-x)y(1-y)\alpha t^{\alpha-1}$. The parameter α gives direct control over the time regularity of the exact solution, which is

$$u^3(x, y, t) = t^\alpha x(1-x)y(1-y).$$

To study the *hp*-DGFEM and the *h*-DGFEM for graded meshes we mainly employ Problem (III) since, with the parameter α, we can generate time singularities as they may occur in real applications. We investigate the performance of the DGFEM by solving the spatial problems (which are smooth) very accurately such that the approximation error which consists of a spatial and a temporal part is clearly dominated by the latter. For the Problems (I) and (II), we employ a uniform grid consisting of 25 spatial elements of uniform approximation order $p = 8$, corresponding to $D = 1681$ spatial dof. The exact solution of Problem (III) is only polynomial of order 2 in x and y such that the spatial discretization with one element of order $p = 2$ is sufficient to represent the exact solution. In the convergence rate plots of Figures 5 through 8 we plot time dof against the relative error in the $L^2(J; H^1(\Omega))$-norm given by $\|u\|_{L^2(J;H^1(\Omega))} = (\int_J \|u(t)\|^2_{H^1(\Omega)} dt)^{\frac{1}{2}}$.

6.2 Performance of the h-Version DGFEM

In the h-version of the DGFEM one chooses a fixed time approximation order r for all time steps of the partition $\mathcal{M}$. Convergence is then achieved by refining the time partition $\mathcal{M}$, i.e., by increasing the number of time steps in the interval $J = (0, T)$. As we have seen in (4), the optimal convergence rates that can be expected in the $L^2(J; H_0^1(\Omega))$-norm are algebraic in the total number of time dof $N = M(r+1)$ with exponent $-\min(r, s) - 1$, cf. Theorem 4. For Problem (I) the left hand side of Figure 5 clearly shows the predicted slopes of $-1, \ldots, -4$ for the approximation orders $r = 0, \ldots, 3$ (with equidistant time steps k). The solution to the Problems (II) and (III) are not arbitrarily smooth in time anymore, and therefore the convergence rates are dominated by the temporal regularity s. But, according to Theorem 11, the optimal convergence rates can be recovered by the use of graded time meshes. We use the grading functions $h(t) = t^{2r+3}$ for Problem (II) and (III), respectively, together with the uniform time approximation order $r = 2$. Figure 6 illustrates that in both cases the optimal slope of -3 is retrieved.

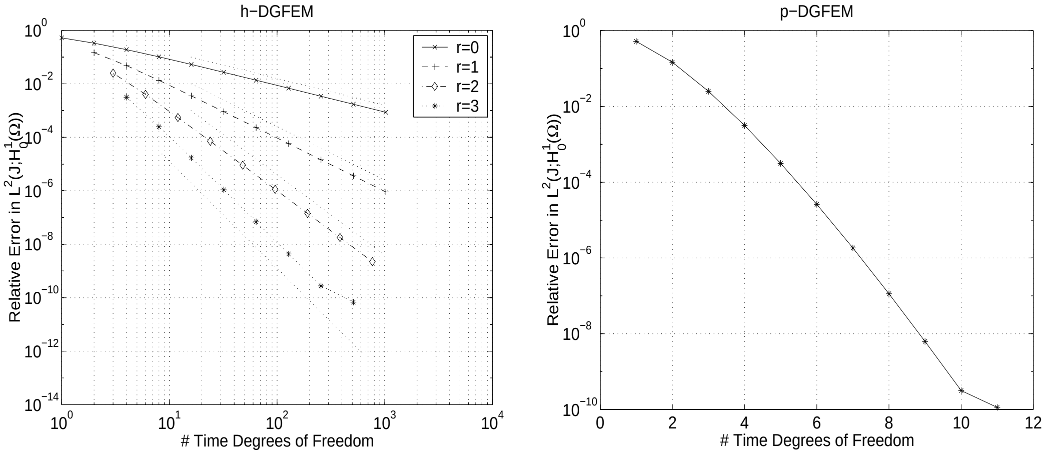

Fig. 5. Convergence rates for Problem (I). Left: h-version DGFEM. The dotted straight lines give the exact slopes $-1, \ldots, -4$. Right: p-version DGFEM.

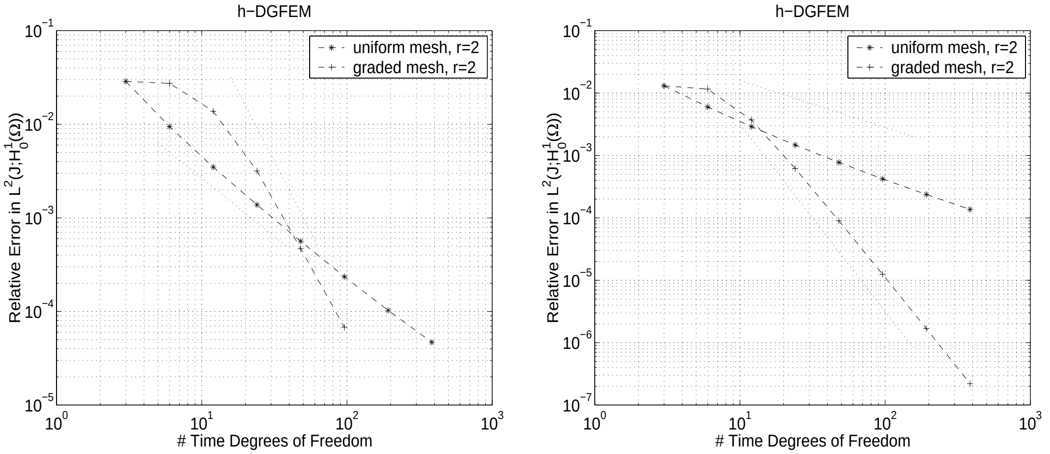

Fig. 6. h-version DGFEM with $r = 2$ on uniform and graded meshes. Left: Problem (II). Right: Problem (III) with $\alpha = 3/4$. In both cases we observe that the optimal convergence rate is recovered for the graded mesh.

6.3 Performance of the p-Version DGFEM

In the p-version of the DGFEM, the time partition $\mathcal{M}$ is fixed. To achieve convergence, one introduces new time dof by increasing the approximation order r uniformly on all time steps. If the exact solution is analytic in time, one can expect exponential convergence rates, cf. Theorem 5. The right side of Figure 5 illustrates this behavior for the smooth Problem (I). Obviously, the p-method is in this case much more efficient than the h-method. As an example: To reduce the relative error to 10^{-6} the p-version needs only 7 time dof while the h-version with $r = 2$ needs approximately 100 time dof. If the exact solution is not analytic in time, the hp-version has to be applied in order to still obtain exponential convergence, as is shown in the following section.

6.4 Performance of the hp-Version DGFEM

In the hp-context of the DGFEM, the time intervals are geometrically refined towards the singularity (which is in our case at the origin of the time axis) while the approximation orders are linearly increased from layer to layer. The two determining parameters for the convergence rates are the geometric grading factor σ and the slope μ of the approximation orders (on time step I_m we have $r_m = \lfloor \mu m \rfloor$). In Figure 7 we show convergence graphs for Problem (III) with $\alpha = 1/2$ and $\alpha = 3/4$, where we set $\sigma = 0.17$ and vary the slope μ. All the graphs show exponential convergence, as predicted in Theorem 9. We note that the optimal slope μ depends on the regularity parameter α. The best choice is approximately: $\mu \approx 1$ for $\alpha = 3/4$ and $\mu \approx 0.75$ for $\alpha = 1/2$. From Figure 8 we can draw two conclusions: On the one hand, the convergence rates are strongly dependent on the grading factor σ (in fact, about two orders of magnitude in the precision are lost if $\sigma = 0.5$ is used instead of the optimal choice of $\sigma \approx 0.17$) and, on the other hand, we can observe that the optimal σ does not depend on α. This is in agreement with [19], where the optimal grading factor was determined to be $\sigma \approx 0.17$ in the context of resolving r^α-singularities for one dimensional boundary value problems.

6.5 CPU-Time Comparison of the Direct versus the Decoupled Method

In this section we present CPU time comparisons to demonstrate the efficiency of the decoupling process described in Section 4.3. The question, how a desired accuracy can be achieved with minimal CPU time is discussed in Section 6.6. Here, we consider the following situation: We have already computed all element matrices for our problem and we therefore have the global mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{S}$ at our disposal. Let us further assume that the

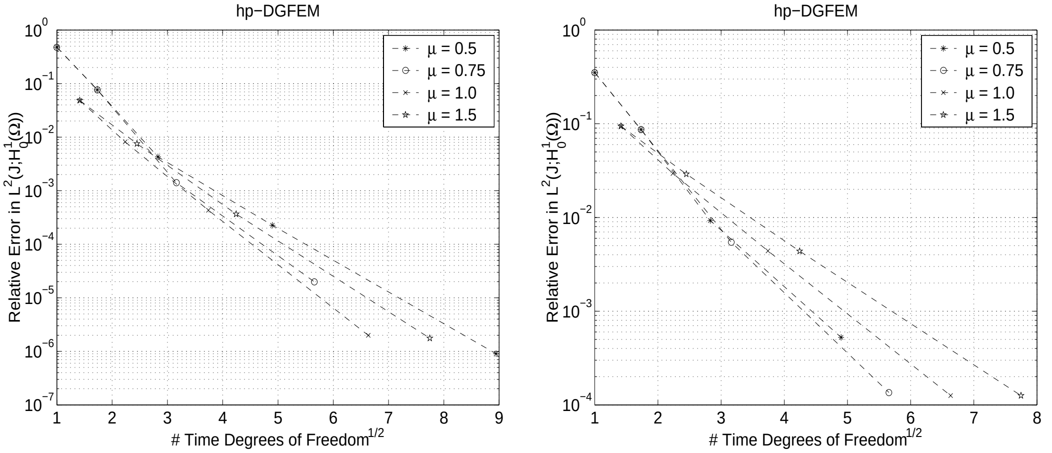

Fig. 7. *hp*-version DGFEM for Problem (III) with grading factor $\sigma = 0.17$ and various μ. Left: $\alpha = 3/4$, optimal $\mu \approx 1$. Right: $\alpha = 1/2$, optimal $\mu \approx 0.75$.

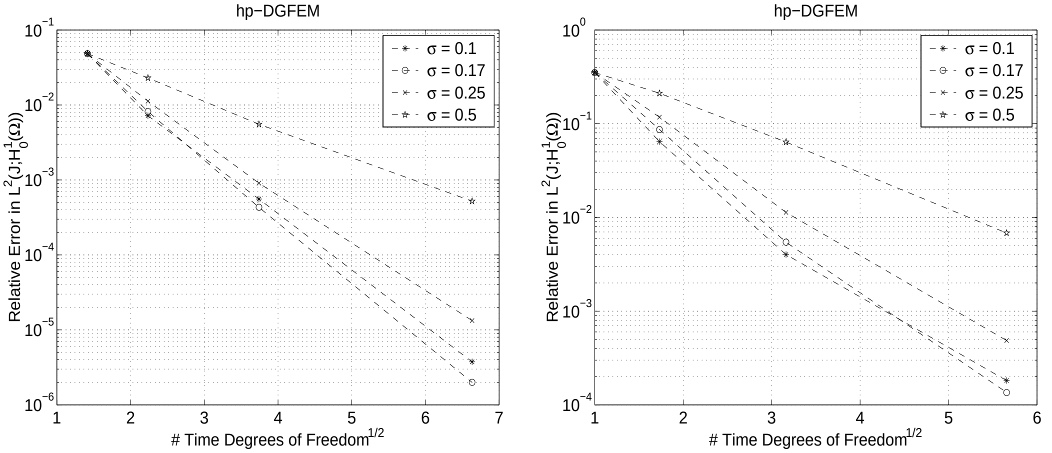

Fig. 8. *hp*-version DGFEM for Problem (III) and various σ. Left: $\alpha = 3/4$, $\mu = 1$. Right: $\alpha = 1/2$, $\mu = 0.75$.

time step size k and/or the time approximation order r are not the same as on the previous time step which typically occurs in the *hp*-DG context and on graded meshes. This implies that we cannot reuse the LU decomposition that we computed for the global system on the previous time step. We have to construct and solve a new system of equations which can be done with various strategies:

- Strategy A : (Full System). Build and solve the block matrix of the coupled system (17) where each block is of the form $\alpha\mathbf{M} + \beta\mathbf{S}$, with $\alpha, \beta \in \mathbb{R}$ (the resulting system matrix is in $\mathbb{R}^{(r+1)D \times (r+1)D}$, D being the number of spatial dof).

- Strategy B : (Decoupled System). Build the $r+1$ matrices $\mathbf{G}_{j,m} = \lambda_j\mathbf{M} + \frac{k}{2}\mathbf{S}$ for the decoupled systems (23). Here, the eigenvalues λ_j are in $\mathbb{C}$ such that the matrices $\mathbf{G}_{j,m}$ are in $\mathbb{C}^{D \times D}$. Additionally, we have to compute the transformed right hand sides $\vec{f}_j^c$ for each system j (compare Example 16). With the $r+1$ solution vectors $\vec{w}_j$ we get the backtransformed solution according to (24).

- Strategy C : (Decoupled, Condensed System). In this case, we cannot reuse the condensed global matrices $\mathbf{M}$ and $\mathbf{S}$. Instead, we have to multiply the element mass and stiffness matrices by the new coefficients λ_j and $\frac{k}{2}$ respectively, condense and reassemble them into $r+1$ system matrices $\mathbf{G}_{j,m} \in \mathbb{C}^{D_{ext} \times D_{ext}}$. The inner dof are computed according to (26). Finally, as in strategy B, the solution has to be backtransformed according to (24).

Remark 20 All global matrices mentioned in the strategies A, B and C are sparse and structurally symmetric. Furthermore, the $r+1$ decoupled systems (23) in the strategies B and C have all the same sparsity pattern. The parallel direct solver PARDISO [28,29] we employ in our code takes advantage of that fact by performing the fill-in reduction and symbolic factorization only once for a certain sparsity pattern.

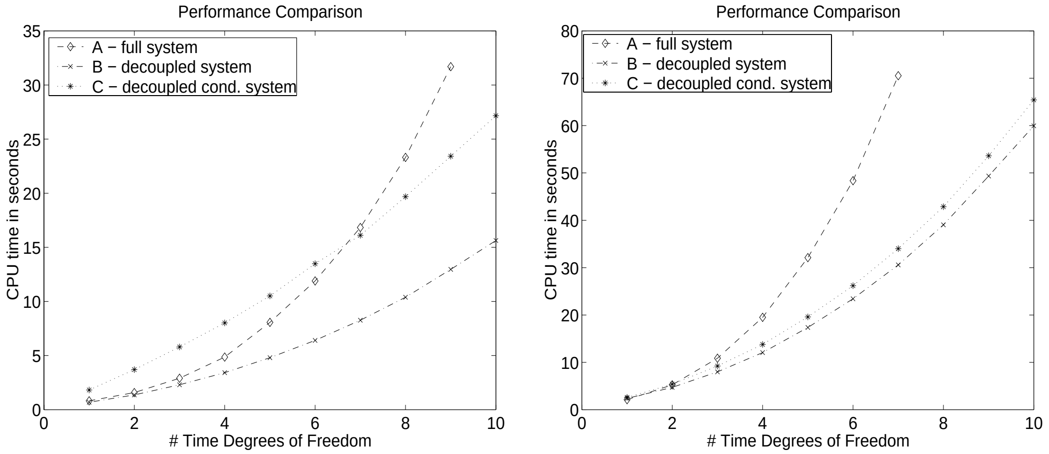

Fig. 9. Time to build and solve the global system on one time step as a function of the time approximation order r. Left: Spatial discretization with 144 quadrilateral elements of uniform approximation order $p = 3$ ($D = 1369$, $D_{ext} = 793$, density $\approx 1.4\%$). Right: 25 elements of uniform approximation order $p = 8$ ($D = 1681$, $D_{ext} = 456$, density $\approx 3.4\%$).

In Figure 9 we give comparisons of the strategies A, B and C in terms of CPU time as a function of the temporal approximation order r. While on the left hand side the spatial approximation is done by 144 quadrilateral elements of uniform approximation order $p = 3$, the problems on the right hand side are discretized with only 25 elements but with a higher approximation order $p = 8$. The difference between the two discretizations for our comparison lies in the sparsity patterns of the matrix $\alpha\mathbf{M} + \beta\mathbf{S}$ (density $\approx 1.4\%$ on the left, density $\approx 3.4\%$ on the right). However, it turns out that in both cases the strategies including decoupling (B and C) are performing best (on the left hand side, strategy A is better than C for $r < 6$, due to the computational overhead for static condensation). Furthermore, we can observe that in this computation it is not worthwhile to do static condensation of the element matrices. However, this conclusion is only valid for the specific problem that we considered. It cannot be generalized since it depends strongly on the spatial discretization (especially on the number of spatial dof D and on the ratio $D_{int} : D_{ext}$ between internal and external dof) and on the performance of the linear solver that is used. If the time step size k and the time approximation order r do *not* change from one time step to the next, then strategy C is the optimal choice, since we only need to compute a back solve for a global system of dimension $D_{ext} \times D_{ext}$. Additionally, in an overall judgement of the effectivity of A, B and C, one should also consider issues such as memory requirements and parallelizability. Including these criteria, strategy C becomes the most attractive one, since the static condensation can be parallelized in a very natural way and reduces the size of the global system considerably, compare Section 4.5.

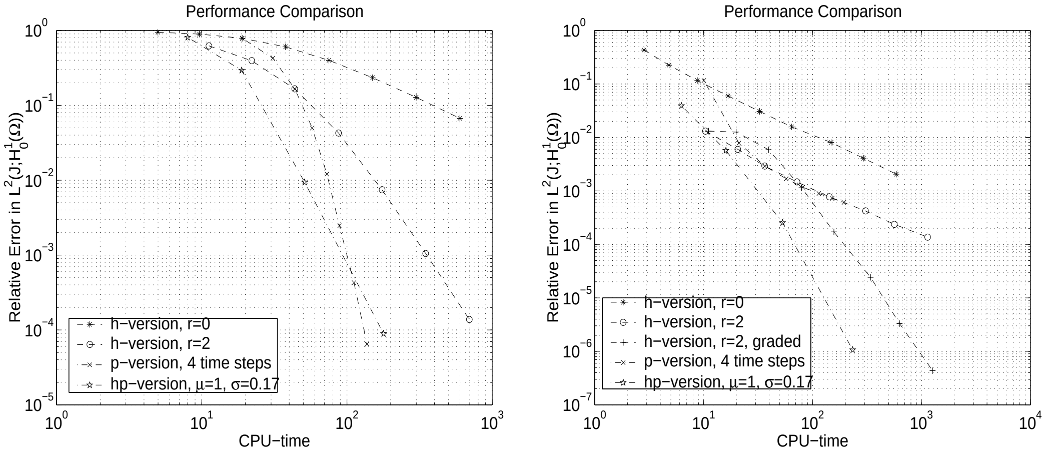

Fig. 10. Relative error vs. CPU-time for different DGFEM strategies. Left: Problem (I), best strategy is the p-version. Right: Problem (III) with $\alpha = 3/4$. Best strategy is the hp-version.

6.6 CPU-Time Comparison of the h-, p- and hp-DGFEM

Up to this point, we were mainly concerned about convergence rates for the different strategies (h-, p- and hp-version) of the DGFEM. But for practical purposes, it is crucial to ask at what cost in terms of CPU time these convergence results can be obtained. To investigate this question experimentally (at the example of Problems (I) and (III) with $\alpha = 3/4$) we plot in Figure 10 the relative error in the $L^2(J; H_0^1(\Omega))$-norm against the CPU time that is needed to solve the problem. Note that the CPU time does not include the time needed for the element integration (it is the same for all strategies), in order to focus our attention on the cost during the time stepping. Considering the results of Section 6.5, all computations are done with the decoupling procedure according to Section 4.3. This allows us to take the total number of time dof N as an estimate for the computational cost of a certain strategy. In fact, we observe in both plots of Figure 10 that the fastest way to obtain a certain precision is the one that needs the least time dof, i.e., the p-version for the smooth Problem (I) and the hp-version for Problem (III). This would be trivial if the time spent to set up and solve one decoupled system was the same for all DGFEM strategies, but this is not the case. As an example: If we choose the h-DG approach, we have to set up $r+1$ linear system and compute their LU factorizations once. In all the following time steps, we only have to perform a backsolve for different right hand sides. On the other hand, in the hp-context (and for graded meshes in the h-context) we have to set up and solve $r+1$ new linear systems in every time step. Still, we can profit from a symbolic factorization that is only determined in the first time step, since all subsequent systems have the same sparsity pattern. However, it turns out that these differences are not significant and that N is in fact an admissible estimate for the computational cost.

We can clearly see that the ability of the hp-DGFEM to resolve start-up singularities is of great value, especially for highly accurate computations. It turns out that the CPU time demand for a h-version DGFEM with a uniform time partition to reach a relative error tolerance of 10^{-5} is significant. In fact, the hp-DGFEM reaches the same error tolerance with several orders of magnitude less CPU time. We conclude that in the presence of start-up singularities the h-DGFEM on graded meshes or the hp-DGFEM are indispensable.

6.7 hp-Scheme in Time and Space for an L-Shaped Domain

In this section we demonstrate the performance of the hp-DGFEM time stepping combined with a continuous hp-FEM approximation in space. We consider again the standard heat equation from Example 1, but this time on the L-shaped domain $\Omega_L \subset \mathbb{R}^2$ shown in Figure 11. An exact solution on $\Omega_L \times J$ is given by (r and θ being the usual polar coordinates)

$$u^L(x,y,t) = t^{\frac{3}{4}} r^{\frac{2}{3}} \sin(\tfrac{2}{3}\theta)(1-x^2)(1-y^2),$$

together with the compatible initial datum $u_0^L = 0$, zero boundary conditions and the right hand side $g^L := \frac{\partial}{\partial t}u^L - \Delta u^L$. We emphasize that we have both a start-up singularity at $t = 0$, and a corner singularity at $r = 0$. These singularities are induced by the right hand side and do not require the use of boundary layers in the spatial mesh. Therefore, we employ geometrically graded meshes in time and space to resolve both singularities. An example of a spatial mesh with $\sigma = 0.15$ and 3 layers of elements is given in Figure 11. In

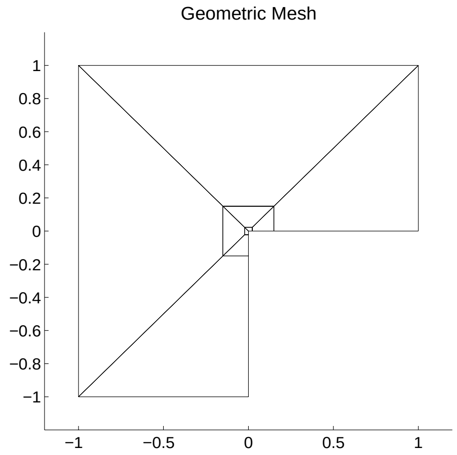

Fig. 11. L-shaped domain Ω_L with 3 layers of spatial elements. The spatial grading factor is $\sigma_s = 0.15$.

Figure 12 we show convergence plots in the $L^2(J; H_0^1(\Omega))$-norm for geometric refinements in time and space, respectively. The sequences of spatial meshes and time meshes that we used to generate these plots are listed in Table 1. First we take the best time discretization of Table 1 and refine geometrically in space. Evidently, the error is then dominated by the spatial error and we obtain exponential convergence with the geometric refinement in space, as is clearly visible on the left side of Figure 12. On the other hand, if we take the best spatial discretization in Table 1, then we obtain exponential convergence due to the geometric refinement in time (grading factor $\sigma = 0.17$ and $\mu = 1$). In conclusion, we can observe exponential convergence for both the time and the spatial discretization.

# Layers	# Elements	p	# Dof
2	6	3	67
3	10	4	181
4	14	5	381
5	18	6	691
6	22	7	1135
7	26	8	1737
8	30	9	2521

# Time Steps	# Time Dof
1	1
2	3
3	6
4	10
5	15
6	21
7	28

Table 1
Discretizations used to generate results in Figure 12. Left: The spatial discretizations. Right: The time discretizations.

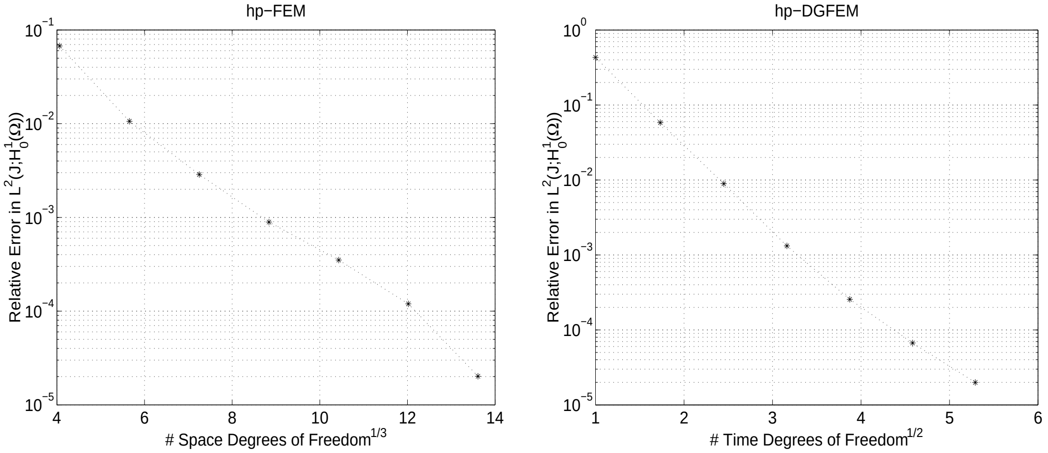

Fig. 12. Convergence rates for the solution u^L. Left: Best time discretization of Table 1, convergence through geometric refinement of the spatial mesh. Right: Best spatial discretization of Table 1, convergence through geometric refinement of the time partition.

7 Summary and Conclusions

In this work, we have described the DGFEM for the time discretization of parabolic problems. The DGFEM is an implicit single time step scheme which allows for arbitrary variations in the time step sizes and the corresponding approximation orders. We have seen that this ansatz leads to a system of equations of dimension $r+1$ on every time step which can be discretized by *hp*-FEM in space in order to obtain a fully discrete *hp*-scheme. Furthermore, we have analyzed a method to decouple this system into $r+1$ independent systems at the price of switching over to complex arithmetic.

In the *hp*-DGFEM time stepping method, we combine geometrically refined time meshes with linearly increasing approximation orders. Theoretical results yield exponential convergence rates even for solutions with time singularities induced by incompatible initial data or piecewise analytic forcing terms. We have applied the *hp*-DGFEM time stepping to the standard heat equation in two dimensional domains and confirmed all predicted convergence rates in our experiments.

In addition, we focussed on algorithmic aspects of the DGFEM and obtained the following conclusions which are crucial for practical purposes:

- The decoupling process is of great value both in terms of computational time and memory requirements. This is true even for low order approximations in space, i.e., for very sparse mass and stiffness matrices.
- The number of time degrees of freedom N that are used to reach a certain relative error is an admissible estimate for the computational cost of the DGFEM. This implies that the exponential convergence rates for the *hp*-DGFEM (in the case of incompatible initial data or piecewise analytic

forcing terms) result directly in a saving of orders of magnitude of CPU time compared to the h-DGFEM.

From the implementational point of view, we have built the hp-DGFEM time stepping on top of the existing code HP90 [10]. The now available extended version of HP90 is a tool to treat parabolic problems by means of hp-FEM in time and space.

Acknowledgements

The author would like to thank the group of W. Fichtner and O. Schenk who made the solver PARDISO available to them.

References

[1] I. Babuška and T. Janik: *The hp-version of the finite element method for parabolic equations, I: The p-version in time, II: The hp-version in time*, Numerical Methods for Partial Differential Equations Vol. 5 (1989), pp. 363-399, and Vol. 6 (1990), pp. 343-369.

[2] I. Babuška and B.Q. Guo: *The hp-version of the finite element method for domains with curved boundaries*, SIAM J. Numer. Anal. Vol. 25 (1988), pp. 837-861.

[3] I. Babuška and M. Suri: *The p- and hp-versions of the finite element method: An overview*, Computer Methods in Applied Mechanics and Engineering Vol. 80 (1990), pp. 5-26.

[4] I. Babuška and M. Suri: *The p- and hp-versions of the finite element method: Basic principles and properties*, SIAM Review Vol. 36 (1994), pp. 578-632.

[5] C.E. Baumann and J.T. Oden: *A discontinuous hp finite element method for convection diffusion problems*, in *Special Issue: Spectral, Spectral Element, and hp methods in CFD*, Ed.: G.E. Karniadakis, Computer Methods in Applied Mechanics and Engineering Vol. 175 (1999), pp. 311-341.

[6] B. Cockburn and C.-W. Shu: *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal. Vol. 35 (1998), pp. 2440-2463.

[7] B. Cockburn: *Discontinuous Galerkin methods for convection dominated problems*, in *High-Order Methods for Computational Physics*, Eds.: T.J. Barth and H. Deconinck, Lecture Notes in Computational Science and Engineering Vol. 9, Springer Verlag, 1999, pp. 69-224.

[8] L. Demkowicz, J.T. Oden, W. Rachowicz and O. Hardy: *Toward a universal hp adaptive finite element strategy, I: Constrained approximation and data structure*, Computer Methods in Applied Mechanics and Engineering Vol. 77 (1989), pp. 79-112.

[9] L. Demkowicz and J.T. Oden: *New developments in applications of hp-adaptive BE/FE methods to elastic scattering*, in *The Mathematics of Finite Elements and Applications*, Ed.: J. R. Whiteman, Wiley, 1994.

[10] L. Demkowicz, K. Gerdes, C. Schwab, A. Bajer and T. Walsh: *HP90: A general and flexible Fortran 90 hp-FE code*, Computing and Visualization in Science, 1998, pp. 145-163.

[11] K. Eriksson and C. Johnson: *Error estimates and automatic time step control for nonlinear parabolic problems*, SIAM J. Numer. Anal. Vol. 24 (1987), pp. 12-23.

[12] K. Eriksson and C. Johnson: *Adaptive finite element methods for parabolic problems, I: A linear model problem*, SIAM J. Numer. Anal. Vol. 28 (1991), pp. 43-77.

[13] K. Eriksson and C. Johnson: *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. Vol. 32 (1995), pp. 706-740.

[14] K. Eriksson and C. Johnson: *Adaptive finite element methods for parabolic problems, III: Time steps variable in space*, in preparation.

[15] K. Eriksson and C. Johnson: *Adaptive finite element methods for parabolic problems, IV: Nonlinear problems*, SIAM J. Numer. Anal. Vol. 32 (1995), pp. 1729-1749.

[16] K. Eriksson and C. Johnson: *Adaptive finite element methods for parabolic problems, V: Long-time integration*, SIAM J. Numer. Anal. Vol.32 (1995), pp. 1750-1763.

[17] K. Eriksson, C. Johnson and S. Larsson: *Adaptive finite element methods for parabolic problems, VI: Analytic semigroups*, SIAM J. Numer. Anal. Vol. 35 (1998), pp. 1315-1325.

[18] K. Eriksson, C. Johnson and V. Thomée: *Time discretization of parabolic problems by the discontinuous Galerkin method*, RAIRO Modél. Math. Anal. Numér. Vol. 19 (1985), pp. 611-643.

[19] W. Gui and I. Babuška: *The h-, p- and hp-versions of the finite element method in one dimension, I: The error analysis of the p-version, II: The error analysis of the h- and hp-versions, III: The adaptive hp-version*, Numer. Math. Vol. 49 (1986), pp. 577-612, 613-657 and 659-683.

[20] B.Q. Guo and I. Babuška: *The hp-version of the finite element method, I: The basic approximation results, II: General results and applications*, Comp. Mech. Vol. 1 (1986), pp. 21-41 and pp. 203-226.

[21] P. LeSaint and P.A. Raviart: *On a finite element method for solving the neutron transport equation*, in *Mathematical Aspects of Finite Elements in Partial Differential Equations*, Ed.: C. de Boor, Academic Press, 1974, pp. 89-145.

[22] J.L. Lions and E. Magenes: *Non-homogeneous Boundary Value Problems and Applications*, Volume I, Springer Verlag, New York, 1972.

[23] C.G. Makridakis and I. Babuška: *On the stability of the discontinuous Galerkin method for the heat equation*, SIAM J. Numer. Anal. 34 (1997), pp. 389-401.

[24] J.M. Melenk and C. Schwab: *hp-FEM for reaction-diffusion equations, I: Robust exponential convergence*, SIAM J. Numer. Anal. 35 (1998), pp. 1520-1557.

[25] J.M. Melenk and C. Schwab: *Analytic regularity for a singularly perturbed problem*, SIAM J. Math. Anal. 30 (1999), pp. 379-400.

[26] J.T. Oden, L. Demkowicz, W. Rachowicz and T.A. Westermann: *Toward a universal hp adaptive finite element strategy, II: A posteriori error estimation*, Computer Methods in Applied Mechanics and Engineering Vol. 77 (1989), pp. 113-180.

[27] J.T. Oden, A. Patra and Y. Feng: *Parallel domain decomposition solver for adaptive hp finite element methods*, SIAM J. Numer. Anal. Vol. 34 (1997), pp. 2090-2118.

[28] O. Schenk, W. Fichtner and K. Gärtner: *Efficient sparse LU factorization with left-right looking strategy on shared memory multiprocessors*, BIT, Vol. 40, No.1, pp.158-176, 1999.

[29] O. Schenk, W. Fichtner and K. Gärtner: *Scalable parallel sparse factorization with left-right looking strategy on shared memory multiprocessors*, In *High-Performance Computing and Networking, No. 1593*, Eds.: P. Sloot, M. Bubak, A. Hoekstra and B. Hertzberger, The Seventh International Conference on High Performance Computing and Networking Europe, April 12-14, 1999, Amsterdam, The Netherlands.

[30] D. Schötzau: *hp-DGFEM for parabolic evolution problems - applications to diffusion and viscous incompressible fluid flow*, Doctoral Dissertation No. 13041, ETH Zürich, 1999.

[31] D. Schötzau and C. Schwab: *Time discretization of parabolic problems by the hp-version of the discontinuous Galerkin finite element method*, SIAM J. Numer. Anal. Vol. 38 (2000), pp. 837-875.

[32] D. Schötzau and C. Schwab: *An hp a-priori error analysis of the DG time-stepping method for initial value problems*, Calcolo Vol. 37 (2000), pp. 207-232.

[33] C. Schwab: *p- and hp-FEM*, Oxford University Press, 1998.

[34] B. Szabó and I. Babuška. *Finite Element Analysis*. Wiley 1991.

[35] V. Thomée: *Galerkin Finite Element Methods for Parabolic Problems*, Springer Verlag, New York, 1997.