hp Discontinuous Galerkin Time Stepping Methods for Parabolic Problems

Klaus Gerdes
Math. Dept., Chalmers University of Technology
SE-412 96 Göteborg, Sweden
email: gerdes@math.chalmers.se

Dominik Schötzau
University of Minneapolis, School of Mathematics
Minneapolis, MN 55455, USA
email: schoetza@math.umn.edu

Christoph Schwab
SAM, ETH Zürich, 8092 Zürich, Switzerland
email: schwab@sam.math.ethz.ch

Thomas Werder
SAM, ETH Zürich, 8092 Zürich, Switzerland
email: twerder@sam.math.ethz.ch

hp DISCONTINUOUS GALERKIN TIME STEPPING FOR PARABOLIC PROBLEMS

KLAUS GERDES, DOMINIK SCHÖTZAU, CHRISTOPH SCHWAB, THOMAS WERDER

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 present 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 condition and the data. Different methods to reduce the computational cost of the DGFEM are compared.

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 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, 12, 13, 14, 15, 16, 17, 18, 23]. We refer also to the recent monograph [35] and the references there. 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 $k \to 0$ and $h \to 0$. Evidently, the convergence mechanism and analysis of the DGFEM time discretization is similar to the one 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 Finite Element Method appeared in the literature (see, e.g., [2, 3, 4, 8, 9, 19, 20, 26, 34] and the references there). 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 in there. The hp-versions of Discontinuous Garlerkin methods have already been applied successfully for the spatial discretization of convection-diffusion problems, we refer to [5, 6, 7] and the references there.

Typically, solutions to parabolic problems are also 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 also be applied in time discretization methods. 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 limits 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 was shown that the hp-DGFEM is able to resolve time singularities at exponential convergence rates, independently of the spatial discretization. The predicted exponential convergence in time and space for parabolic equations was confirmed for one dimensional model problems in time and space [30, 31]. We also mention [32] where new a-priori estimates are 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 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 in detail the predicted theoretical results on exponential convergence. Furthermore, CPU time comparisons for the h, p and hp-DGFEM show clearly 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]. In particular, it is predicted that in the presence of incompatible initial data, exponential convergence can be achieved by the use of the hp-DGFEM. The implementational aspects of the DGFEM are discussed in Section 4. Section 5 contains a short discussion of possible parallelization strategies. And, finally, in Section 6 we present numerical results and CPU time measurements obtained for the different versions of the DGFEM applied to the two dimensional heat equation.

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]. $H_0^1(\Omega)$ 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 $r \leq r_m$ with coefficients in X. $C_0^\infty(I; X)$ 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 for Parabolic Problems

We review the theoretical setting for the *hp*-DGFEM time semidiscretization from [30, 31]. The spatial discretization is adressed in Section 4.

2.1. **The Abstract Parabolic Problem.**

We consider general linear parabolic problems of the form

$$u'(t) + Lu(t) = g(t), \qquad t \in J = (0, T), \tag{2.1}$$

$$u(0) = u_0. \tag{2.2}$$

where L is assumed to be an elliptic spatial operator, u_0 the initial data and g the forcing term. The precise functional framework we use is as follows. Let X and H be complex, separable Hilbert spaces with dense injection $X \overset{d}{\hookrightarrow} H$ and norms $\|\cdot\|_X$ and $\|\cdot\|_H$. We assume in addition that $X \overset{d}{\hookrightarrow} H$ is compactly imbedded. The operator L is given as $(Lu, v)_H = a(u, v)$, where $(\cdot,\cdot)_H$ denotes the inner product on H and $a : X \times X \to \mathbb{R}$ is a continuous, coercive sesquilinearform, i.e.,

$$|a(u, v)| \leq \alpha \|u\|_X \|v\|_X \qquad \forall u, v \in X, \tag{2.3}$$

$$\operatorname{Re} a(u, u) \geq \beta \|u\|_X^2 \qquad \forall u \in X, \tag{2.4}$$

$$a(u, v) = \overline{a(u, v)} \qquad \forall u, v \in X. \tag{2.5}$$

It is always assumed that the initial condition u_0 satisfies $u_0 \in H$, and that the forcing term satisifies $g \in L^2(J; H)$. The weak formulation of the problem (2.1)-(2.2) is obtained by multiplying (2.1) with test functions $v \in X$ and $\varphi(t) \in C_0^\infty(J)$ followed by an appropriate integration by parts.

Definition 2.1. *(Weak Form). 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*

$$-\int_J (u(t), v)_H \varphi'(t) dt + \int_J a(u, v)\varphi(t) dt = \int_J (g(t), v)_{X^* \times X} \varphi(t) dt \tag{2.6}$$

for all $v \in X$ and $\varphi \in C_0^\infty(J)$. Here, X^* denotes the dual space of X.

Under the assumptions $u_0 \in H$ and $g \in L^2(J; H)$ there exists a unique weak solution to (2.6), see, e.g., [22].

2.2. **DGFEM Time Semidiscretization.** We first semidiscretize the parabolic problem (2.1)-(2.2) in time by the DGFEM, the space discretization is adressed in Section 4. We show that for the linear problem in (2.1)-(2.2) the DGFEM results in an unconditionally stable, implicit single step scheme where arbitrary variations in the time steps and the approximation orders are allowed.

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 $0 =: t_0 < t_1 < \ldots < t_{M-1} < t_M := T$. We also introduce a vector $\underline{r}$ which specifies the temporal approximation order r_m on each time step. An illustration of a time mesh can be found in Figure 2.1.

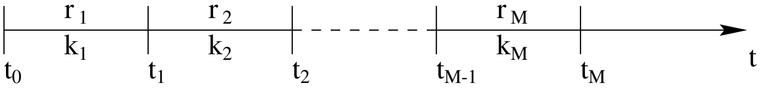

FIGURE 2.1. (Time Mesh $\mathcal{M}$). The length of time step I_m is $k_m := t_m - t_{m-1}$ for $1 \le m \le M$. The time approximation order on time step I_m is r_m.

The idea of the DGFEM is to approximate the exact solution u by a semidiscrete function U which consists on every time step of a polynomial in the time variable t of order r_m and with coefficients in X. The polynomials on the different time steps are not required to be continuous across the time nodes. This allows, cf. (2.11) ahead, to write the DGFEM as a time stepping scheme. In order to deal with the discontinuities we need the following definitions.

Definition 2.2. *(One sided limits, jumps). The left and right handed limits of a function $u : J \to H$ (or $u : J \to X$) at time node t_m are defined to be*

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

Furthermore, the jump across time node t_m is defined as

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

For the exact solution $u \in L^2(J; X) \cap H^1(J; X^*)$ these limits exist in H. In Figure 2.2 we give an illustration of Definition 2.2.

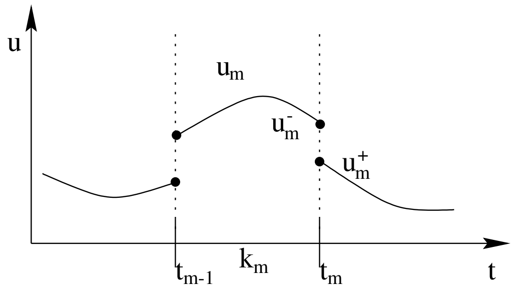

FIGURE 2.2. Symbolic sketch of the one sided limits u_m^+, u_m^- and of the jump $[u]_m$ across the time node t_m.

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

$$C_b(\mathcal{M};X) := \{u : J \to X \text{ with } u|_{I_m} \in C_b(I_m;X)\} \tag{2.7}$$

consisting of bounded continuous functions on each time step. We define the bilinear form B_{DG} and the linear form F_{DG} by

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

Then it is straightforward to see by integration by parts that the following Lemma holds:

Lemma 2.3. *Let $u \in L^2(J;X) \cap H^1(J;X^*)$ be a weak solution of (2.1)-(2.2) in the sense of (2.6). 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 semidiscrete 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

$$\mathcal{V}^{\underline{r}}(\mathcal{M};X) := \{u : J \to X : u|_{I_m} \in \mathcal{P}^{r_m}(I_m;X) \text{ for } 1 \le m \le M\}. \tag{2.8}$$

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 dof $N = \text{NRDOF}(\mathcal{V}^{\underline{r}}(\mathcal{M};X))$ can be considered as a crude measure for the computational cost of the DGFEM.

Definition 2.4. *(DGFEM). Let $\mathcal{M}$ be a partition of the time interval $J = (0,T)$ and $\underline{r}$ an approximation order distribution on $\mathcal{M}$. The DGFEM applied to the parabolic problem (2.1)-(2.2) is to:*

(2.9)
$$\textit{Find } U \in \mathcal{V}^{\underline{r}}(\mathcal{M};X) \textit{ such that } \quad B_{DG}(U,V) = F_{DG}(V) \quad \forall\, V \in \mathcal{V}^{\underline{r}}(\mathcal{M};X).$$

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

Proposition 2.5. *The DGFEM (2.9) has a unique solution $U \in \mathcal{V}^{\underline{r}}(\mathcal{M};X)$. If u is the exact solution of (2.1)-(2.2), we have the Galerkin orthogonality*

$$B_{DG}(u - U, V) = 0 \qquad \forall V \in \mathcal{V}^{\underline{r}}(\mathcal{M};X). \tag{2.10}$$

Due to the discontinuity of the test and trial spaces $\mathcal{V}^{\underline{r}}(\mathcal{M};X)$, the DGFEM (2.9) can be interpreted as an implicit time stepping scheme. This means that the DGFEM solution U can be found by solving successively for $m = 1, \ldots, M$ the following problems: Find U_m such that

$$\begin{aligned} &\int_{I_m} \{(U',V)_H + a(U,V)\} dt + (U_{m-1}^{+}, V_{m-1}^{+})_H \\ &\qquad = \int_{I_m} (g,V)_{X^* \times X} dt + (U_{m-1}^{-}, V_{m-1}^{+})_H. \end{aligned} \tag{2.11}$$

for all $V \in \mathcal{P}^{r_m}(I_m; X)$. U_{m-1}^- corresponds to the initial condition in time step I_m (here, 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.

In Section 4 we choose a concrete basis $\{\varphi_{i,m}\}_{i=0}^r$ for the space $\mathcal{P}^{r_m}(I_m; X)$ and describe more algorithmic aspects of the DGFEM. Additionally, we introduce a spatial FEM discretization to obtain a fully discrete *hp*-scheme, but first we give an error analysis of the time semidiscretization in the next section.

3. A Priori Error Estimates for the Time Discretization

We cite some a priori estimates for the DG time stepping method which are derived in [30, 31]. See also [32] for an error analysis of the DGFEM time discretization applied to initial value problems.

3.1. *hp* Approximation Results.

Theorem 3.1. *Let $u \in H^{s+1}(J; X)$ for $s \geq 0$, let $r_m = r$ on each time interval I_m, and set $k = \max\{k_m\}$. For the DG solution $U \in \mathcal{V}^r(\mathcal{M}; X)$ 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 consider two important applications of Theorem 3.1. 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 results:

- 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

$$\|u - U\|_{L^2(J;X)} \leq C N^{-\min(r,s)-1}. \tag{3.1}$$

- 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

$$\|u - U\|_{L^2(J;X)} \leq C N^{-s-1}. \tag{3.2}$$

By inspection of the two latter estimates, we can see that the application of the *p*-version is advantageous if the exact solution is smooth. 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 3.2. *Let the exact solution u 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 3.1 and Theorem 3.2 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 $H_\theta = (H, X)_{\theta,2}$, $0 \le \theta \le 1$, defined by the K-method of interpolation. The parameter θ measures the compatibility of the initial datum u_0 with respect to a continuous scale of intermediate spaces $X \subseteq H$. 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 3.3. *Let the right hand side g in (2.1) be piecewise analytic (in time) and let $u_0 \in H_\theta$ for some $0 \le \theta \le 1$. Then the solution u of (2.1)-(2.2) satisfies*

$$\|u^{(l)}(t)\|_X^2 \le Cd^{2l}\Gamma(2l+2)t^{-(2l+1)+\theta}.$$

in the vicinity of $t = 0$.

Theorem 3.3 gives a precise characterization of the start-up singularity induced by incompatible initial data. Results of this type have been known, see [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 3.3 by the *hp*-version DGFEM. For simplicity, we confine ourselves to the time interval $J = (0, 1)$.

Definition 3.4. *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*

$$t_0 = 0, \qquad t_m = \sigma^{n-m+1}, \qquad 1 \le m \le n+1. \tag{3.3}$$

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

Definition 3.5. *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 $1 \le m \le n+1$.*

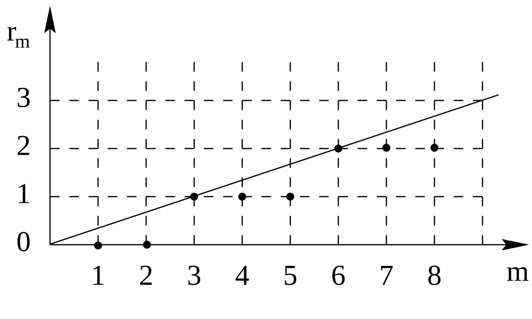

FIGURE 3.1. (Linear polynomial degree vector). Example with slope $\mu = 1/3$ on 8 time steps. The resulting degree vector is $\underline{r} = (0,0,1,1,1,2,2,2)$.

Figure 3.1 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 for temporal singularities as in Theorem 3.3. The following theorem gives a precise formulation of this fact.

Theorem 3.6. *Consider the parabolic problem (2.1)-(2.2) on $J = (0,1)$ with initial value $u_0 \in H_\theta$ 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 $\mathcal{M}_\sigma$ 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*

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

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 (3.1). In the Finite Element Method 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 $J = (0,1)$.

Definition 3.7. *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*

$$h \in C^0([0,1]) \cap C^1((0,1)), \qquad h(0) = 0, \qquad h(1) = 1. \tag{3.4}$$

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

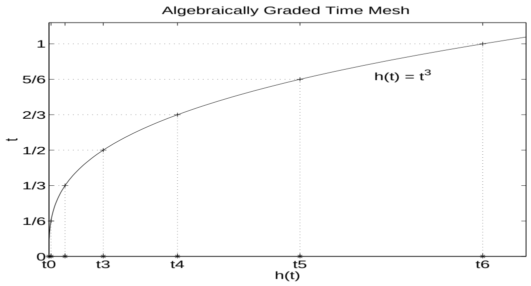

FIGURE 3.2. (Algebraically Graded Time Mesh $\mathcal{M}$). Example for 6 time steps in the interval $J = (0,1)$ with grading function $h(t) = t^3$.

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 3.8 states that we can recover the optimal h-version convergence rate by the use of an appropriate grading function h.

Theorem 3.8. *Let u be the solution of (2.1)-(2.2) with $u_0 \in H_\theta$ 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 $U \in \mathcal{V}^r(\mathcal{M}; X)$*

$$\|u - U\|_{L^2(J;X)} \leq CM^{-(r+1)} \qquad \textit{or} \qquad \|u - U\|_{L^2(J;X)} \leq CN^{-(r+1)} \tag{3.5}$$

with C depending only on u_0, g and r.

4. The Fully Discrete *hp* Scheme

In this section we focus on the fully discrete *hp*-scheme, i.e., in addition to the *hp*-DGFEM in time we introduce a *hp*-FEM for the spatial discretization.

We define a basis of time shape functions $\{\varphi_{i,m}(t)\}_{i=0}^{r_m}$ for $\mathcal{P}^{r_m}(I_m;X)$ and write for the semidiscrete solution U_m on every time step I_m

$$U_m = \sum_{j=0}^{r_m} u_{j,m}\varphi_{j,m}, \tag{4.1}$$

with $u_{j,m} \in X$. We show that with (4.1) the DGFEM amounts to a system of r_m+1 coupled elliptic reaction diffusion equations for the coefficients $u_{j,m}$ in space. In order to obtain a fully discrete solution these equations have to be discretized in space by a Finite Element Method. To do so, we introduce a finite element space $X_D \subset X$ of dimension $\dim(X_D) = D$ and determine FE approximatons $u_{j,m}^{FE}$ to the coefficients $u_{j,m}$. On timestep I_m the fully discrete solution is then of the form

$$U_m^{FE} = \sum_{j=0}^{r_m} u_{j,m}^{FE}\varphi_{j,m}. \tag{4.2}$$

4.1. Semidiscretization in Time.

We focus on the solution of the problem (2.11) on one time step $I_m = (t_{m-1}, t_m)$ with time approximation order r_m. The initial condition U_{m-1}^- as well as the right hand side $g(t)$ are given.

We introduce a basis $\{\widehat{\varphi}_i\}_{i=0}^{r}$ of reference time shape functions for the polynomial space $\mathcal{P}^r((-1,1))$. The time shape functions $\varphi_{i,m}(t)$ on the interval (t_{m-1}, t_m) are then defined as $\varphi_{i,m} \circ F_m(\hat{t}) = \widehat{\varphi}_i(\hat{t})$, where the corresponding mapping function F_m from the reference interval to (t_{m-1}, t_m) is

$$t = F_m(\hat{t}) := \frac{1}{2}(t_{m-1} + t_m) + \frac{1}{2}\hat{t}k_m. \tag{4.3}$$

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

$$U_m = \sum_{j=0}^{r_m} u_{j,m}\varphi_{j,m}, \qquad V_m = \sum_{i=0}^{r_m} v_{i,m}\varphi_{i,m}. \tag{4.4}$$

We choose normalized Legendre polynomials as reference time shape functions, that is

$$\widehat{\varphi}_j(\hat{t}) = \sqrt{(j+1/2)} \cdot L_j(\hat{t}), \quad j \geq 0, \tag{4.5}$$

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

Example 4.1. *With (4.5) the first five reference time shape functions are given by*

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

In order to avoid cumbersome notation, we consider for the rest of this section a generic time step $I = (t_0, t_1)$ and omit the time step index m on the time shape functions $\varphi_{i,m}$, on the coefficients $u_{j,m}$ and $v_{i,m}$, on the mapping F_m, as well as on the approximation order r_m. By inserting the ansatz (4.4) in the time stepping scheme (2.11) we obtain the following system of equations:

Find coefficients $\{u_j\}_{j=0}^r \subset X$ such that

$$\sum_{i,j=0}^{r}\{\overbrace{[\int_I \varphi_j'\varphi_i dt + \varphi_j^+(t_0)\varphi_i(t_0)]}^{A_{ij}}(u_j,v_i)_H + \overbrace{[\int_I \varphi_j\varphi_i dt]}^{B_{ij}}\, a(u_j,v_i)\} \tag{4.6}$$
$$= \sum_{i=0}^{r}\{\overbrace{(\int_I g\varphi_i dt, v_i)_H}^{f_i^1} + \overbrace{(U_0^-, v_i)_H\varphi_i^+(t_0)}^{f_i^2}\} \qquad \text{for all} \quad \{v_i\}_{i=0}^r \subset X.$$

Using the mapping F in (4.3), we transform problem (4.6) to a problem on the reference interval $(-1,1)$. For clarity, we introduce the following abbreviations in terms of the time reference shape functions $\widehat{\varphi}_i(\hat t)$:

$$\hat A_{ij} := \int_{-1}^{1} \hat\varphi_j'\hat\varphi_i d\hat t + \hat\varphi_j^+(-1)\hat\varphi_i^+(-1), \qquad \hat B_{ij} := \int_{-1}^{1} \hat\varphi_j\hat\varphi_i d\hat t, \tag{4.7}$$

$$\hat f_i^1(v) := (\hat l_i^1, v)_H, \qquad \hat l_i^1 := \int_{-1}^{1} (g\circ F)\widehat{\varphi}_i d\hat t, \tag{4.8}$$

$$\hat f_i^2(v) := (\hat l_i^2, v)_H, \qquad \hat l_i^2 := U_0^-\widehat{\varphi}_i(-1). \tag{4.9}$$

Using these abbreviations, we can write the system (4.6) in the compact form:

Find coefficients $\{u_j\}_{j=0}^r \subset X$ such that for all $\{v_i\}_{i=0}^r \subset X$

$$\sum_{i,j=0}^{r} \hat A_{ij}(u_j,v_i)_H + \frac{k}{2}\hat B_{ij} a(u_j,v_i) = \sum_{i=0}^{r} \frac{k}{2}\hat f_i^1(v_i) + \hat f_i^2(v_i). \tag{4.10}$$

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 data U_0^-. Note that on the first time step we have $U_0^- = u_0$.

Remark 4.2. *The matrices* $\hat{\mathbf{A}}$ *and* $\hat{\mathbf{B}}$ *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 (4.12) 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 4.3. *For the time shape functions in (4.5) 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*

$$\hat{\mathbf{A}} = \begin{pmatrix} 0.5000 & 0.8660 & 1.1180 & 1.3228 & 1.5000 & 1.6583 \\ -0.8660 & 1.5000 & 1.9365 & 2.2913 & 2.5981 & 2.8723 \\ 1.1180 & -1.9365 & 2.5000 & 2.9580 & 3.3541 & 3.7081 \\ -1.3229 & 2.2913 & -2.9580 & 3.5000 & 3.9686 & 4.3875 \\ 1.5000 & -2.5981 & 3.3541 & -3.9686 & 4.5000 & 4.9749 \\ -1.6583 & 2.8723 & -3.7081 & 4.3875 & -4.9749 & 5.5000 \end{pmatrix}. \tag{4.11}$$

Note that 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 (4.11) containing the first $r+1$ rows and columns, e.g., for $r = 2$ we have

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

Mathematically, (4.10) represents the variational formulation of an elliptic system of $r+1$ equations whose strong form is

$$\sum_{j=0}^{r} \hat{A}_{ij} u_j + \frac{k}{2} \delta_{ij} L u_j = \frac{k}{2} \hat{l}_i^1 + \hat{l}_i^2, \qquad i = 0, \ldots, r. \tag{4.12}$$

Equivalently, we obtain in matrix notation

$$\hat{\mathbf{A}} \vec{u} + \frac{k}{2} [\delta_{ij}] L \vec{u} \quad = \quad \frac{k}{2} \vec{l}^1 + \vec{l}^2, \tag{4.13}$$

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

Remark 4.4. *For the time approximation order $r = 0$, the DGFEM is 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 (4.10) in space (for details about *hp*-FEM theory, we refer to [33]). We choose a finite dimensional subspace $X_D \subset X$ of dimension $D = \dim(X_D)$ and look for finite element approximations $u_j^{FE} \in X_D$ to u_j. 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

$$u_j^{FE} = \sum_{l=1}^{D} u_j^l s_l(x), \qquad v_i^{FE} = \sum_{k=1}^{D} v_i^k s_k(x). \tag{4.14}$$

We insert this ansatz into (4.10) and define the mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{S}$ to be

$$\mathbf{M} := \{(s_l, s_k)_H\}_{l,k=1}^{D}, \qquad \mathbf{S} := \{a(s_l, s_k)\}_{l,k=1}^{D}. \tag{4.15}$$

The fully discrete system that we get 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

$$\begin{bmatrix} \hat{A}_{00}\mathbf{M} + \frac{k}{2}\mathbf{S} & \cdots & \hat{A}_{0r}\mathbf{M} \\ \vdots & \ddots & \vdots \\ \hat{A}_{r0}\mathbf{M} & \cdots & \hat{A}_{rr}\mathbf{M} + \frac{k}{2}\mathbf{S} \end{bmatrix} \begin{bmatrix} \vec{u}_0 \\ \vdots \\ \vec{u}_r \end{bmatrix} = \frac{k}{2} \begin{bmatrix} \vec{f}_0^1 \\ \vdots \\ \vec{f}_r^1 \end{bmatrix} + \begin{bmatrix} \vec{f}_0^2 \\ \vdots \\ \vec{f}_r^2 \end{bmatrix} \tag{4.16}$$

with load vectors

$$\vec{f}_j^1 = (\hat{f}_j^1(s_1), \hat{f}_j^1(s_2), \ldots, \hat{f}_j^1(s_D))^T, \tag{4.17}$$

$$\vec{f}_j^2 = (\hat{f}_j^2(s_1), \hat{f}_j^2(s_2), \ldots, \hat{f}_j^2(s_D))^T. \tag{4.18}$$

Note that a linear system of the above type (4.16) 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 costly to solve the fully discrete system (4.16). Therefore, it would be interesting to decouple the system (4.13) into $r+1$ scalar problems, that could be solved independently. However, this seems not to be possible with time shape functions in $\mathbb{R}$, but numerical experiments show that the matrix $\hat{\mathbf{A}}$ in (4.7), evaluated for the Legendre time shape functions, is diagonalizable in $\mathbb{C}$ at least for $0 \le r \le 100$: There exists a matrix $\mathbf{Q} \in \mathbb{C}^{(r+1)\times(r+1)}$ such that

$$\mathbf{Q}^{-1}\hat{\mathbf{A}}\mathbf{Q} = \hat{\mathbf{T}} = \mathrm{diag}(\lambda_1^{(r)}, ..., \lambda_{r+1}^{(r)}), \tag{4.19}$$

with pairwise complex conjugate eigenvalues $\lambda_j = \lambda_j^{(r)}$. These matrices allow us to decouple the system (4.13) in the following way

$$\underbrace{\mathbf{Q}^{-1}\hat{\mathbf{A}}\mathbf{Q}}_{\hat{\mathbf{T}}}\underbrace{\mathbf{Q}^{-1}\vec{u}}_{\vec{w}} + \frac{k}{2}L\underbrace{\mathbf{Q}^{-1}\vec{u}}_{\vec{w}} = \mathbf{Q}^{-1}(\frac{k}{2}\vec{l}^1 + \vec{l}^2), \tag{4.20}$$

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

$$\hat{\mathbf{T}}\vec{w} + \frac{k}{2}L\vec{w} = \mathbf{Q}^{-1}(\frac{k}{2}\vec{l}^1 + \vec{l}^2). \tag{4.21}$$

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

$$\lambda_j w_j + \frac{k}{2}L w_j = [\frac{k}{2}\mathbf{Q}^{-1}\vec{l}^1 + \mathbf{Q}^{-1}\vec{l}^2]_j \quad \text{for } 0 \le j \le r. \tag{4.22}$$

In practical applications we limit ourselves to time approximation orders of $r \le 12 = r_{max}$. The corresponding matrices $\mathbf{Q}$, $\mathbf{Q}^{-1}$ and $\hat{\mathbf{T}}$, for $0 \le r \le r_{max}$, 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 (4.22) in $\mathbb{C}$. In this section we consider the spatial discretization of these systems by *hp*-FEM techniques. The standard weak formulation of the equations in (4.22) is

$$\begin{aligned} &\text{Find } w_j \in X \text{ such that for all } v \in X: \\ &b_j(w_j, v) := \lambda_j(w_j, v)_H + \frac{k}{2}a(w_j, v) = \hat{f}_j^c(v), \end{aligned} \tag{4.23}$$

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

$$\hat{f}^c_j = [\frac{k}{2} Q^{-1} \vec{\hat{f}}^1 + Q^{-1} \vec{\hat{f}}^2]_j, \tag{4.24}$$

with

$$\vec{\hat{f}}^i = (\hat{f}^i_0, \hat{f}^i_1, ..., \hat{f}^i_r)^T, \quad i = 1, 2. \tag{4.25}$$

The corresponding FEM approximation is

$$\text{Find } w^{FE}_j \in X_D \quad \text{such that} \qquad b_j(w^{FE}_j, v) = \hat{f}^c_j(v) \qquad \text{for all} \quad v \in X_D. \tag{4.26}$$

Inserting for w^{FE}_j an ansatz of type (4.14) in (4.26) yields the following linear system for the unknown coefficient vector $\vec{w}_j = (w^1_j, w^2_j, .., w^D_j)^T \in \mathbb{C}^D$

$$\overbrace{[\lambda_j \mathbf{M} + \frac{k}{2}\mathbf{S}]}^{\mathbf{G}_j}\, \vec{w}_j = \vec{\hat{f}}^c_j \tag{4.27}$$

with $\vec{\hat{f}}^c_j = (\hat{f}^c_j(s_1), \hat{f}^c_j(s_2), ..., \hat{f}^c_j(s_D))^T$. We call the matrix $\mathbf{G}_j := \lambda_j \mathbf{M} + \frac{k}{2}\mathbf{S}$ the global matrix of the specific j-th spatial system in (4.22) with the mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{S}$ defined as in (4.15). We get the coefficients $\vec{u}_j$ for the functions u^{FE}_j by applying the backtransformation

$$\vec{u}_j = \sum_{i=1}^{r+1} Q_{ji} \vec{w}_j. \tag{4.28}$$

We give the following example to clarify the structure of the load vector $\vec{\hat{f}}^c_j$ in (4.27).

Example 4.5. *Let the time approximation order be $r = 1$, then the loadvector $\vec{\hat{f}}^c_j$ 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.

Remark 4.6. *Each of the decoupled equations (4.22) corresponds to the singularly perturbed model problem*

$$\varepsilon^2 L w + w = f \tag{4.29}$$

where $\varepsilon = \varepsilon^{(r)}_{j,m} = \sqrt{k_m/(2\lambda_j)} \in \mathbb{C}$, with $\lambda_j = \lambda^{(r)}_j$ of (4.19), $\sqrt{\cdot}$ being the usual principal branch of the square root taken to be positive on $(0, \infty)$.

In the following Lemma 4.7 of [31] which analyzes the dependence of λ_j (and therefore also ε) on r, we see that the modulus $|\varepsilon|$ can approach zero.

Lemma 4.7. *Let $\lambda_j \in \mathbb{C}$ be an eigenvalue of the matrix $\hat{A}$ in (4.7). Then*

$$\operatorname{Re} \lambda_j \leq C_1 \max(1, r^2) \quad \text{and} \quad 0 < C_2 \leq |\lambda_j| \leq C_3 \max(1, r^2) \tag{4.30}$$

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

The small parameter ε in 4.29 causes difficulties due to the appearance of boundary layers. In [24, 25] it is shown that the hp-version FEM for problem (4.29) 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 interal dof by local static condensation. To do so, the dof in all element matrices $\mathbf{E}^{[k]} = \alpha\mathbf{M}^{[k]} + \beta\mathbf{S}^{[k]}$, $\forall k$, 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

$$\mathbf{E}_c = \mathbf{E}_{ee} - \mathbf{E}_{ei}\overbrace{\mathbf{E}_{ii}^{-1}\mathbf{E}_{ie}}^{\mathbf{C}}, \qquad \vec{f}_c = \vec{f}_e - \mathbf{E}_{ei}\overbrace{\mathbf{E}_{ii}^{-1}\vec{f}_i}^{\vec{c}}. \tag{4.31}$$

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:

$$\vec{x}_i = \mathbf{E}_{ii}^{-1}\vec{f}_i - \mathbf{E}_{ii}^{-1}\mathbf{E}_{ie}\vec{x}_e. \tag{4.32}$$

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 (4.32) 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 describe now the fully discrete DGFEM time stepping algorithm in the case where we decouple the spatial systems (4.12). 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 (4.27) and can therefore be computed in advance as well.

Algorithm 4.8. *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 is described in Algorithm 4.9 with $ElemM(:)$, $MassM(:)$, $StiffM(:)$ and $LoadV(:)$ being arrays of element matrices and element load vectors respectively.

Algorithm 4.9. *DGFEM Time Stepping, decoupled, sequential*

```
  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 spatial discretization (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 $k = 1, r_m$
          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
```

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 $j = 0, \dots, r_m$ 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 $\mathbf{G}_{j,m}$, $j = 0, \dots, r$, 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.

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 (4.27) 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 the solver *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 fulfils $q \gg r+1$ 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 which describes the temperature field $u(x,y,t)$ in an isotropic body $\Omega \subset \mathbb{R}^2$ (with Dirichlet boundary conditions, i.e., $u|_{\partial\Omega} = 0$) in the time interval $J = (0,T)$ with an initial temperature distribution given by $u_0(x,y)$:

$$\frac{\partial}{\partial t}u - \Delta u = g \quad \text{in } \Omega \times J, \tag{6.1}$$

$$u = 0 \quad \text{on } \partial\Omega \times J, \tag{6.2}$$

$$u|_{t=0} = u_0 \quad \text{in } \Omega. \tag{6.3}$$

To cast (6.1)-(6.3) in the framework presented in Section 2.1, we have to set

$$X = H_0^1(\Omega), \qquad H = L^2(\Omega), \tag{6.4}$$

$$a(u,v) = \int_\Omega \nabla u \nabla v \; dxdy, \qquad f(v) = \int_\Omega gv \; dxdy. \tag{6.5}$$

In the following computations, we take the computational domain to be the unit square $\Omega = (0,1)^2$ and the time interval $J = (0, 0.1)$. We only integrate to the time $t_M = 0.1$ because the solutions that we consider are strongly smoothend 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 (1):
  - Initial solution : $u_0^1(x,y) = \sin(\pi x)\sin(\pi y)$.
  - Right hand side : $g = 0$.
  - u_0^1 is actually the first eigenfunction of the Laplacian and compatible with being in $H_0^1(\Omega)$. The corresponding exact solution $u^1(x,y,t)$ is smooth in space and time:

$$u^1(x,y,t) = \exp(-2\pi^2 t)\sin(\pi x)\sin(\pi y). \tag{6.6}$$

- Problem (2):
  - Initial solution : $u_0^2(x,y) = x(1-x)y(1-y)$.
  - Right hand side : $g = 0$.
  - u_0^2 is also compatible with being in $H_0^1(\Omega)$. The exact solution $u^2(x,t)$ is represented as a Fourier series with coefficients a_{kl}^2:

$$u^2(x,t) = \sum_{l=1}^{\infty}\sum_{k=1}^{\infty} a_{kl}^2 \exp(-\pi^2(l^2+k^2)t)\sin(l\pi x)\sin(k\pi y), \tag{6.7}$$

$$a_{kl}^2 = 16\frac{(1-\cos(l\pi))(1-\cos(k\pi))}{l^3k^3\pi^6}. \tag{6.8}$$

- Problem (3)
  - Initial solution: $u_0^3(x,y) = x(1-x)y(1-y)$.
  - Right hand side: $g = 2t^\alpha(x(1-x)+y(1-y)) + x(1-x)y(1-y)\alpha t^{\alpha-1}$
  - The initial condition u_0^3 is in $H_0^1(\Omega)$. The parameter α gives direct control over the time regularity of the exact solution, which is

$$u^3(x,y,t) = t^\alpha x(1-x)y(1-y). \tag{6.9}$$

To study the *hp*-DGFEM and the *h*-DGFEM for graded meshes we mainly employ Problem (3) 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 (1) and (2), 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 (3) 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 6.1 - 6.4 we plot time dof against the relative error in the $L^2(J;H_0^1(\Omega))$-norm:

$$\|u\|_{L^2(J;H_0^1(\Omega))} = \left(\int_J \|u(t)\|^2_{H_0^1(\Omega)}dt\right)^{\frac{1}{2}}. \tag{6.10}$$

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 (3.1), 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 3.1. For Problem (1) the left hand side of Figure 6.1 clearly shows the predicted slopes of $-1, \dots, -4$ for the approximation orders $r = 0, \dots, 3$ (with equidistant time steps k). The solution to the Problems (2) and (3) are not arbitrarly smooth in time, and therefore the convergence rates are dominated by the temporal regularity s. But, as stated in Theorem 3.8, 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 (2) and (3) respectively, together with the uniform time approximation order $r = 2$. Figure 6.2 illustrates that in both cases the optimal slope of -3 is retrieved.

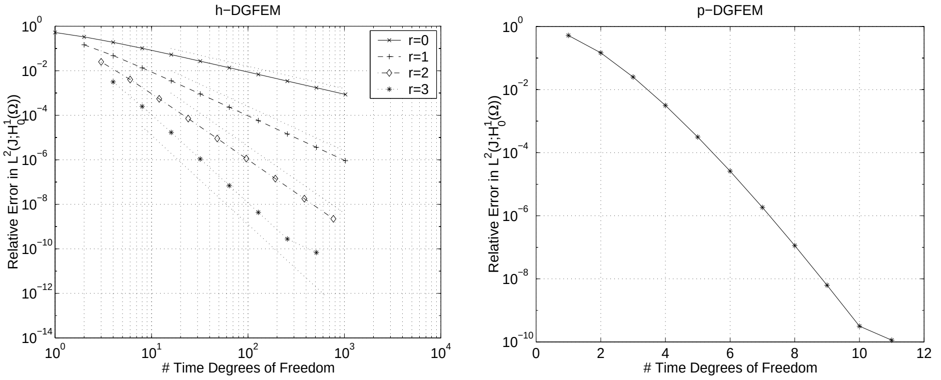

FIGURE 6.1. Convergence rates for Problem (1). Left: h-version DGFEM. The dotted straight lines give the exact slopes $-1, \dots, -4$. Right: p-version DGFEM.

6.3. **Performance of the p-Version DGFEM.** In the p-version of the DGFEM, the time partion $\mathcal{M}$ 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 3.2. The right side of Figure 6.1 illustrates this behaviour for the smooth Problem (1). 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 obtain optimal results (exponential convergence), as is shown in the following section.

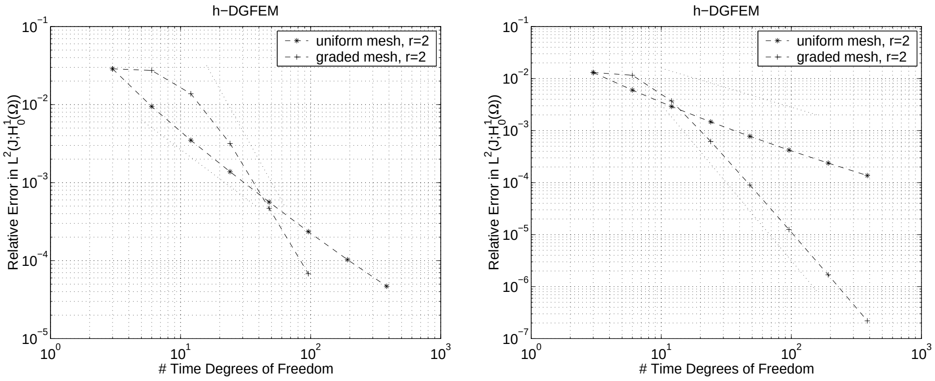

FIGURE 6.2. h-version DGFEM with $r = 2$ on uniform and graded meshes. Left: Problem (2). Right: Problem (3) with $\alpha = 3/4$. In both cases we observe that the optimal convergence rate is recovered for the graded mesh.

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 geometrical 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 Fig 6.3 we show convergence graphs for Problem (3) 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 3.6. 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 6.4 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 problems in space.

6.5. **CPU-Time Comparison of the Direct vs. 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 disposition. Let us further assume that the 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:

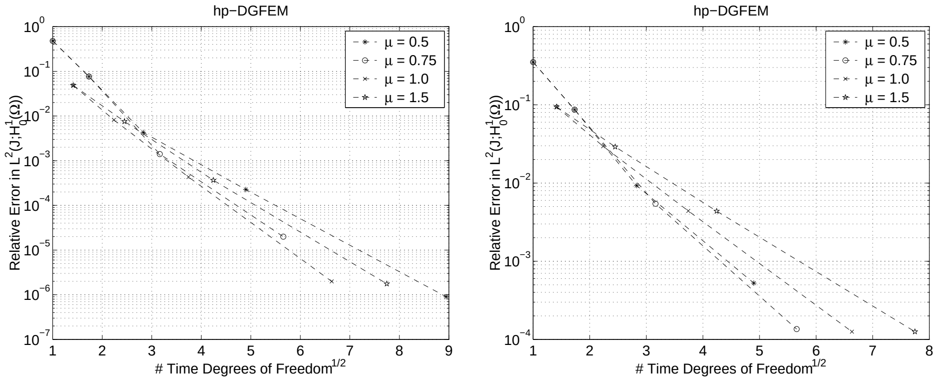

Figure 6.3. *hp*-version DGFEM for Problem (3) with grading factor $\sigma = 0.17$ and various μ. Left: $\alpha = 3/4$, optimal $\mu \approx 1$. Right: $\alpha = 1/2$, optimal $\mu \approx 0.75$.

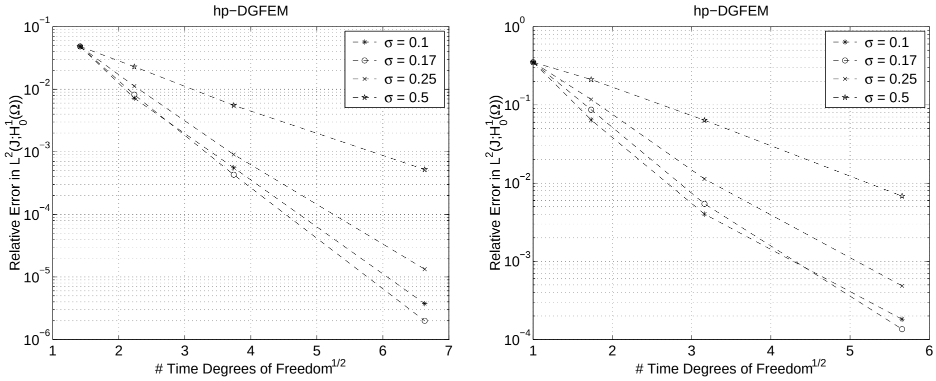

Figure 6.4. *hp*-version DGFEM for Problem (3) and various σ. Left: $\alpha = 3/4$, $\mu = 1$. Right: $\alpha = 1/2$, $\mu = 0.75$.

- Strategy A : (Full System). Build and solve the block matrix of the coupled system (4.16) 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 (4.27). 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 $\vec{f}^c_j$ for each system j (compare Example 4.5). With the $r+1$ solution vectors $\vec{w}_j$ we get the back transformed solution according to (4.28).

- 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 (4.32). Finally, as in strategy B, the solution has to be backtransformed according to (4.28).

Remark 6.1. *All global matrices mentioned in the strategies A, B and C are sparse and structurally symmetric. Furthermore, the* $r+1$ *decoupled systems (4.27) in the strategies B and C have all the same sparsity pattern. The parallel direct solver PARDISO* [28, 29]*, that 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.*

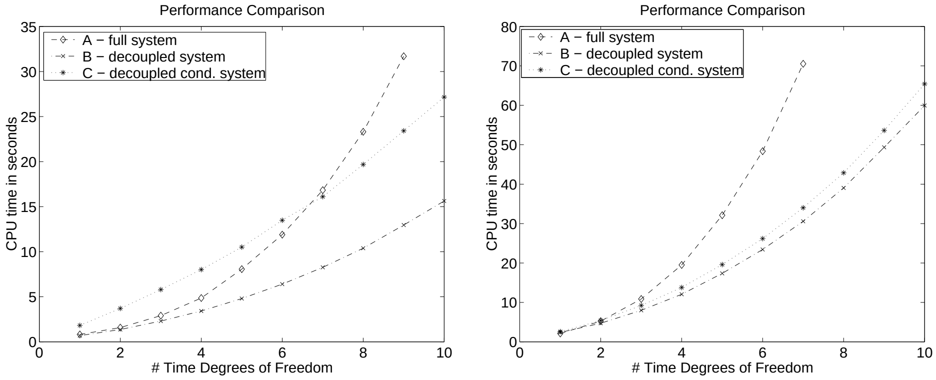

FIGURE 6.5. Time to build and solve the global system on one time step as a function of the time approximation order r. Left: Spatial discretization with 144 quadrilateral elements of uniform approximation order $p = 3$ ($D = 1369$, $D_{ext} = 793$, density $\approx$ 1.4%). Right: 25 elements of uniform approximation order $p = 8$ ($D = 1681$, $D_{ext} = 456$, density $\approx$ 3.4%).

In Fig. 6.5 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), on the performance of the linear solver that is used and on the implementation. 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.

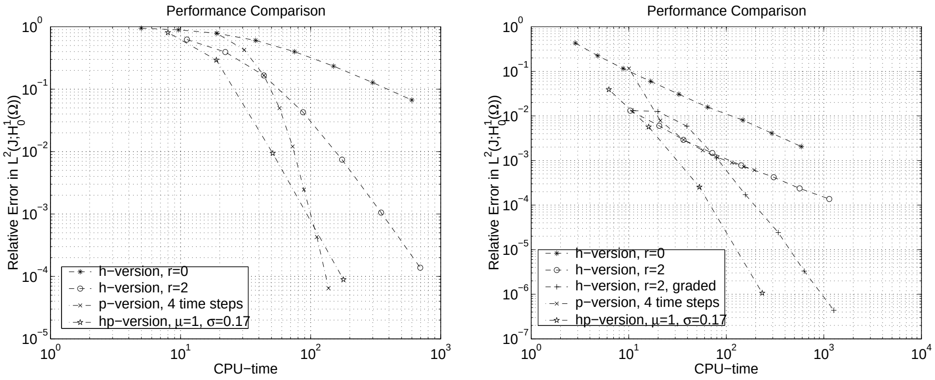

FIGURE 6.6. Relative error vs. CPU-time for different DGFEM strategies. Left: Problem (1), best strategy is the p-version. Right: Problem (3) with $\alpha = 3/4$. Best strategy is the hp-version.

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, hp) 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 the Problems (1) and (3) with $\alpha = 3/4$) we plot in Figure 6.6 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 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 Fig. 6.6 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 (1) and the hp-version for Problem (3). 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 on 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.

# 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. Meshes used to generate Figure 6.8. Left: The spatial discretizations. Right: The time discretizations.

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 (6.1)-(6.3) but this time on the L-shaped domain $\Omega_L \subset \mathbb{R}^2$ of Figure 6.7. An exact solution is given on $\Omega_L \times J$ (r and θ being the usual polar coordinates) by

$$u^L(x,y,t) = t^{\frac{3}{4}} r^{\frac{2}{3}} \sin(\tfrac{2}{3}\theta)(1-x^2)(1-y^2) \tag{6.11}$$

together with the compatible initial condition $u_0^L = 0$, zero boundary conditions and the right hand side

$$g^L := \frac{\partial}{\partial t} u^L - \Delta u^L. \tag{6.12}$$

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 6.7. In Figure 6.8 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 6.8. 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.

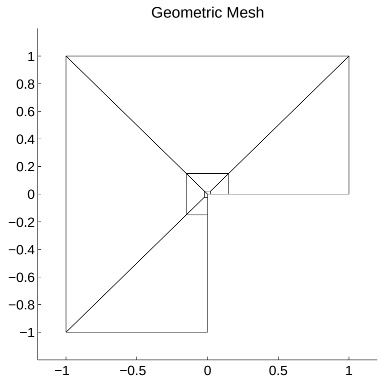

FIGURE 6.7. L-shaped domain Ω_L with 3 layers of spatial elements. The spatial grading factor is $\sigma_s = 0.15$.

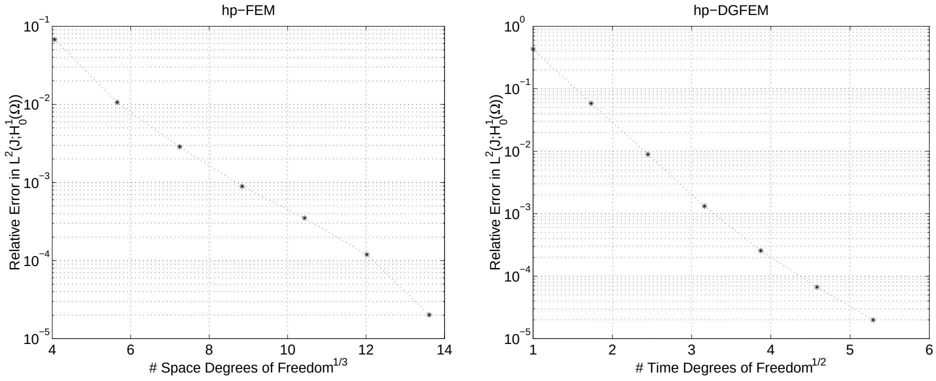

FIGURE 6.8. Convergence rates for Problem (4). Left: Best time discretization of Table 1, convergence through geometrical refinement of the spatial mesh. Right: Best spatial discretization of Table 1, convergence through geometrical refinement of the time partition.

7. CONCLUSIONS & SUMMARY

In this work we have described the DGFEM for the time semidiscretization of parabolic problems. The DGFEM is an implicit single time step scheme which allows for arbitray variations in the time step sizes and the corresponding approximation orders. The main idea of the method is to seek an approximation that is on every time step a polynomial in time of order r. These polynomials have not necessarily to be continuous across the time steps. We have seen that this ansatz leads to a system of equations of dimension $r+1$ on every time step which has to be discretized by a *hp*-FEM in space 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. We have encountered three different possibilities to achieve convergence with the DGFEM: In the h-version we reduce the size of the time steps uniformly at a fixed approximation order r, in the

p-version we keep a fixed time partition but increase the approximation orders, and finally, in the hp-version, we combine the two approaches in an optimal way, i.e., we construct geometrically refined time meshes with linearly increasing approximation orders. A theoretical analysis predicts exponential convergence for the p-version, if the exact solution is arbitrarly smooth in time, and for the hp-version, if we deal with incompatible initial data or a piecewise analytic forcing term.

We have applied the hp-DGFEM time stepping to the standard heat equation in a two dimensional domain Ω and confirmed all predicted convergence rates. 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 built the hp-DGFEM time stepping on top of the existing code HP90 [10]. This code was developed to solve general elliptic problems in a hp-FEM context and has been applied to problems such as the Poisson equation, the Stokes problem, and linear elasticity. The now available extended version of HP90 is a tool to treat the time dependent extensions of these problems by means of hp-FEM in time and space. As a first example, we considered the standard heat equation on an L-shaped domain Ω_L. Despite a corner singularity in space and a start up singularity in time we achieved exponential convergence as predicted by the theory.

8. Acknowledgements

The kind support of Chalmers Tekniska Högskola for this research project is greatfully acknowledged. A special thank goes to the group of W. Fichtner and O. Schenk who made the solver PARDISO available.

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*, Computer Methods in Applied Mechanics and Engineering (1999), to appear.

[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. Part 1: Constrained Approximation and Data Structure. *Computer Methods in Applied Mechanics and Engineering*, 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. by 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 I*, 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*, Numerische Mathematik 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, Part I: The basic approximation results, Part 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] Ch.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. Part 2: A Posteriori Error Estimation. *Computer Methods in Applied Mechanics and Engineering*, 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*, Technical Report 98/40, Integrated Systems Laboratory, ETH Zürich, Submitted to BIT Numerical Mathematics, 1998.

[29] O. Schenk, W. Fichtner and K. Gärtner, *Scalable Parallel Sparse Factorization with Left-Right Looking Strategy on Shared Memory Multiprocessors*, In P. Sloot, M. Bubak, A. Hoekstra, and B. Hertzberger (Eds.), High-Performance Computing and Networking, No. 1593, pp. 221-230, Springer-Verlag. 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*, SAM Report 99-04, Seminar für Angewandte Mathematik, ETH Zürich, 1999, submitted to SIAM J. Num. Anal.

[32] D. Schötzau and C. Schwab: *An hp a-priori error analysis of the DG time-stepping method for initial value problems*, IMA Research Report 1652:1999, University of Minneapolis, accepted in Calcolo.

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

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

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