SIAM J. Sci. Comput.
Vol. 30, No. 1, pp. 369–393

ADAPTIVITY WITH DYNAMIC MESHES FOR SPACE-TIME FINITE ELEMENT DISCRETIZATIONS OF PARABOLIC EQUATIONS[*]

MICHAEL SCHMICH[†] AND BORIS VEXLER[‡]

Abstract. In this paper, we develop an error estimator and an adaptive algorithm for efficient solution of parabolic partial differential equations. The error estimator assesses the discretization error with respect to a given quantity of physical interest and separates the influence of the time and space discretizations. This allows us to set up an efficient adaptive strategy producing economical (locally) refined meshes for each time step and an adapted time discretization. The space and time discretization errors are equilibrated, leading to an efficient method.

Key words. parabolic equations, a posteriori error estimation, mesh refinement, space-time finite elements, dynamic meshes

AMS subject classifications. 65N30, 65N50, 65M50, 35K55

DOI. 10.1137/060670468

1. Introduction.

In this paper, we develop an adaptive algorithm for efficient solution of (nonlinear) parabolic equations and systems of them. Both time and space discretizations of the underlying equations are based on finite element methods as proposed, e.g., in [10, 12]. Within this adaptive algorithm, the time discretization and the space discretizations for each time step are successively refined in order to reduce the overall discretization error in an efficient way. To this end, we derive an a posteriori error estimator which assesses the error between the solution of the continuous and the discretized problems with respect to a given quantity of interest describing the goal of the computation. The presented error estimator separates the influence of time and space discretization on the error in the quantity of interest. This allows for balancing these two types of error. Moreover, this error estimator provides information for separate refinement of the finite element meshes corresponding to different time steps. The use of such *dynamic meshes* leads to very efficient numerical schemes and is described in this paper.

The use of adaptive techniques based on a posteriori error estimation is well accepted in the context of finite element discretizations of partial differential equations; see, e.g., [9, 19, 4]. Different adaptive algorithms for parabolic problems are discussed, e.g., in [13, 15, 17, 6, 1]. In these publications, the adaptivity is based on error estimation with respect to some global norms. However, in many applications, the error in global norms does not provide useful error bounds for the error in the quantity of physical interest. In [3, 4], a general concept for a posteriori estimation of the discretization error with respect to a given functional is presented. In a recent paper [16], a posteriori error estimates are derived for parabolic optimization problems.

In this paper, we extend the approach from [4] to parabolic equations and derive an a posteriori error estimator allowing for efficient use of dynamic locally refined

[*] Received by the editors September 22, 2006; accepted for publication (in revised form) June 21, 2007; published electronically January 4, 2008.
http://www.siam.org/journals/sisc/30-1/67046.html

[†] Institut für Angewandte Mathematik, Ruprecht-Karls-Universität Heidelberg, Im Neuenheimer Feld 294, 69120 Heidelberg, Germany (michael.schmich@iwr.uni-heidelberg.de).

[‡] Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, Altenberger Straße 69, 4040 Linz, Austria (boris.vexler@oeaw.ac.at).

meshes and nonuniform time discretizations. The evaluation of the presented error estimator is based on the solution of an appropriate dual (adjoint) problem. This dual problem corresponds to a linearization of the (nonlinear) parabolic equation and has to be solved backwards in time. The corresponding additional numerical costs are discussed in what follows.

To the best of the authors' knowledge, this is the first paper providing an adaptive algorithm which allows for equilibrating space and time discretization errors for computation on dynamic meshes. This is possible due to the structure of the error estimator which provides quantitative information about both the time and the space discretization errors.

In this paper, we consider parabolic problems

$$\begin{aligned} \partial_t u + A(u) &= F, \\ u(0) &= u^0, \end{aligned}$$

with a (nonlinear) elliptic operator A and an initial condition u^0. A variational formulation of this problem is presented using the following abstract setting: Let V and H be Hilbert spaces which build together with the dual space V^* of V a Gelfand triple $V \hookrightarrow H \hookrightarrow V^*$. The inner product of H is denoted by $(\cdot,\cdot)_H$. A typical choice for these spaces could be

$$V = \left\{ v \in H^1(\Omega) \,\middle|\, v\big|_{\Gamma_\mathrm{D}} = 0 \right\} \text{ and } H = L^2(\Omega), \tag{1.1}$$

where Γ_D denotes the part of the boundary $\partial\Omega$ of the computational domain $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) with prescribed Dirichlet boundary conditions.

For a time interval $I = (0, T)$, we introduce the Hilbert space $X := W(0, T)$ defined as

$$W(0,T) := \left\{ v \,\middle|\, v \in L^2(I, V) \text{ and } \partial_t v \in L^2(I, V^*) \right\}.$$

It is well known that the space X is continuously embedded in $C(\bar{I}, H)$; see, e.g., [8]. Furthermore, we use the inner product of $L^2(I, H)$ given by

$$(u, v) := (u, v)_{L^2(I,H)} = \int_0^T (u(t), v(t))_H \, dt. \tag{1.2}$$

For a given spatial semilinear form $\bar{a}\colon V \times V \to \mathbb{R}$ describing a weak formulation of a (nonlinear) elliptic operator A, we define the (time-dependent) semilinear form $a\colon X \times X \to \mathbb{R}$ by

$$a(u)(\varphi) := \int_0^T \bar{a}(u(t))(\varphi(t)) \, dt.$$

The corresponding weak formulation of the considered problem has the following form: Find $u \in X$ such that

$$\begin{aligned} (\partial_t u, \varphi) + a(u)(\varphi) &= (f, \varphi) \quad \forall \varphi \in X, \\ u(0) &= u^0, \end{aligned} \tag{1.3}$$

where $f \in L^2(0, T; V^*)$ represents the (time-dependent) right-hand side and $u^0 \in H$ describes initial conditions.

Furthermore, we consider a functional $J\colon X \to \mathbb{R}$ representing the quantity of physical interest. This functional is given as a sum:

$$J(u) := \int_0^T J_1(u(t))\,dt + J_2(u(T)), \tag{1.4}$$

where $J_1\colon V \to \mathbb{R}$ and $J_2\colon H \to \mathbb{R}$ are three times continuously differentiable functionals. This choice covers the following two typical situations: The quantity of interest is a mean value of a given functional (J_1), or one is interested in a terminal value $J_2(u(T))$.

Let u_{kh} be a solution of the discretized version of (1.3), and then we aim at the a posteriori error estimation with respect to J of the following type:

$$J(u) - J(u_{kh}) \approx \eta_k + \eta_h,$$

where η_k describes the error due to the time discretization and η_h the error due to the discretization in space.

Let us briefly outline the following sections. In section 2, we describe the space-time finite element discretization of (1.3). Moreover, we present some details of computing on dynamic (varying in time) meshes. Section 3 is devoted to the derivation of a posteriori error estimates for the discretization error with respect to the quantity of interest J. The error estimates assess separately the error due to the time and to the space discretizations and are obtained by using the solution of a (linear) adjoint equation. In section 4, we describe the numerical evaluation of the derived error estimates and an adaptive algorithm for successive improvement of the accuracy. In section 5, numerical examples for a nonlinear combustion model are presented, illustrating the behavior of the method.

2. Discretization.

In this section, we describe the space-time finite element discretization of (1.3). The space discretization will be based on usual H^1-conforming finite elements, and for the discretization in time we present the following two concepts: *discontinuous Galerkin* (dG) and *continuous Galerkin* (cG) methods. For both methods, the test functions will be discontinuous in time. The trial functions are discontinuous in the dG setting, whereas the cG method uses continuous trial functions.

In the following subsection, we discuss the semidiscretization in time using these concepts. Subsection 2.2 deals with the space discretization of the arising semidiscrete problems.

2.1. Time discretization.

Let

$$\bar{I} = \{0\} \cup I_1 \cup \cdots \cup I_M$$

be a partition of the time interval $\bar{I} = [0,T]$ into subintervals $I_m := (t_{m-1}, t_m]$ of size $k_m := t_m - t_{m-1}$ with given time points

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

The discretization parameter k is defined as a piecewise constant function by $k\big|_{I_m} = k_m$ for $m = 1, \ldots, M$. By means of the subintervals I_m, we define for $r \in \mathbb{N}_0$ two semidiscrete spaces X_k^r and $\widetilde{X}_k^r$:

$$\begin{aligned}
X_k^r &:= \left\{\, v_k \in C(\bar{I}, H) \,\middle|\, v_k\big|_{I_m} \in \mathcal{P}_r(I_m, V) \,\right\} \subset X,\\
\widetilde{X}_k^r &:= \left\{\, v_k \in L^2(I, V) \,\middle|\, v_k\big|_{I_m} \in \mathcal{P}_r(I_m, V) \text{ and } v_k(0) \in H \,\right\}.
\end{aligned}$$

Here $\mathcal{P}_r(I_m, V)$ denotes the space of polynomials up to degree r on I_m with values in V. Thus, X_k^r consists of piecewise polynomials which are continuous in time and will be used as the trial space for the continuous Galerkin method. The functions in $\widetilde{X}_k^r$ may have discontinuities at the ends of the subintervals I_m. This space will be used as the test space for the continuous Galerkin method and as the trial and test space for the discontinuous Galerkin method.

For setting up the dG(r) formulation, we define for a function $v_k \in \widetilde{X}_k^r$:

$$v_{k,m}^{\pm} := \lim_{\varepsilon \downarrow 0} v_k(t_m \pm \varepsilon), \quad [v_k]_m := v_{k,m}^+ - v_{k,m}^-.$$

Then the dG(r) formulation of problem (1.3) reads: Find $u_k \in \widetilde{X}_k^r$ such that

$$\begin{aligned}(2.1)\qquad \sum_{m=1}^{M} \int_{I_m} (\partial_t u_k, \varphi)_H \, dt + a(u_k)(\varphi) + \sum_{m=0}^{M-1} ([u_k]_m, \varphi_m^+)_H + (u_{k,0}^-, \varphi_0^-)_H \\ = (f, \varphi) + (u^0, \varphi_0^-)_H \quad \forall \varphi \in \widetilde{X}_k^r.\end{aligned}$$

Using the semidiscrete spaces defined above, the cG(r) ($r \geq 1$) formulation can be directly stated as: Find $u_k \in X_k^r$ such that

$$(2.2)\qquad (\partial_t u_k, \varphi) + a(u_k)(\varphi) + (u_k(0), \varphi_0^-)_H = (f, \varphi) + (u^0, \varphi_0^-)_H \quad \forall \varphi \in \widetilde{X}_k^{r-1}.$$

Due to the fact that the test space in both formulations is discontinuous in time, these discretizations result in time-stepping schemes. In the next section, we discuss the resulting formulation for the low order methods dG(0) and cG(1) combined with a space discretization.

2.2. Space discretization. In this section, we describe the Galerkin finite element discretization in space of the arising semidiscrete problems. To this end, we use two- or three-dimensional shape-regular meshes; see, e.g., [7]. A mesh consists of quadrilateral or hexahedral cells K which conform a nonoverlapping cover of the computational domain $\Omega \subset \mathbb{R}^d$. The corresponding mesh is denoted by $\mathcal{T}_h = \{K\}$, where the discretization parameter h is defined as a cellwise constant function by setting $h\big|_K = h_K$, with the diameter h_K of the cell K.

Remark 2.1. In order to ease mesh refinement, we allow for cells to have nodes which lie on midpoints of faces of the neighboring cells. But at most one such *hanging node* is permitted for each cell. There are no degrees of freedom corresponding to these irregular nodes, and the value of a finite element function is determined by pointwise interpolation.

On the mesh $\mathcal{T}_h$, we construct a conforming finite element space $V_h^s \subset V$ in a standard way:

$$V_h^s := \left\{ v \in V \,\middle|\, v\big|_K \in \mathcal{Q}_s(K) \text{ for } K \in \mathcal{T}_h \right\}.$$

We use isoparametric elements; i.e., $\mathcal{Q}_s(K)$ consists of shape functions obtained via $\widehat{\mathcal{Q}_s}(\widehat{K})^d$ transformations of polynomials in $\widehat{\mathcal{Q}_s}(\widehat{K})$ defined on the reference cell $\widehat{K} = (0,1)^d$, where

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

Remark 2.2. The definition of V_h^s can be extended to the case of triangular meshes in the obvious way.

To obtain the fully discrete versions of the time discretized problems (2.1) and (2.2), we allow dynamic mesh change in time, but the time steps k_m are kept constant in space. To this end, we associate with each time point t_m a mesh $\mathcal{T}_h^m$ and a corresponding (spatial) finite element space $V_h^{s,m}$. We then define the following space-time finite element space:

$$\widetilde{X}_{k,h}^{r,s} := \left\{ v_{kh} \in L^2(I,V) \,\middle|\, v_{kh}\big|_{I_m} \in \mathcal{P}_r(I_m, V_h^{s,m}) \text{ and } v_{kh}(0) \in V_h^{s,0} \right\} \subset \widetilde{X}_k^r.$$

Then the so-called cG(s)dG(r) formulation of problem (1.3) reads: Find $u_{kh} \in \widetilde{X}_{k,h}^{r,s}$ such that

$$(2.3)\quad \sum_{m=1}^{M} \int_{I_m} (\partial_t u_{kh}, \varphi)_H \, dt + a(u_{kh})(\varphi) + \sum_{m=0}^{M-1} ([u_{kh}]_m, \varphi_m^+)_H + (u_{kh,0}^-, \varphi_0^-)_H$$
$$= (f,\varphi) + (u^0, \varphi_0^-)_H \quad \forall \varphi \in \widetilde{X}_{k,h}^{r,s}.$$

Remark 2.3. The notation cG(s)dG(r) is taken from [11] and describes a method with conforming (continuous) discretization in space of order s and discontinuous discretization in time of order r.

The formulation of the corresponding cG(s)cG(r) discretization is more involved since we have to ensure the global continuity of functions in the trial space. To this end, we describe an approach similar to the one presented in [5]. Let $\{\tau_0, \dots, \tau_r\}$ be a basis of $\mathcal{P}_r(I_m, \mathbb{R})$ with the following property:

$$\tau_0(t_{m-1}) = 1, \quad \tau_0(t_m) = 0, \quad \tau_i(t_{m-1}) = 0, \ i = 1, \dots, r.$$

We set

$$(2.4)\quad X_{k,h}^{r,s,m} := \operatorname{span}\left\{ \tau_i v_i \,\middle|\, v_0 \in V_h^{s,m-1}, \ v_i \in V_h^{s,m}, \ i = 1, \dots, r \right\} \subset \mathcal{P}_r(I_m, V)$$

and define the trial space for the cG(s)cG(r) formulation by

$$X_{k,h}^{r,s} := \left\{ v_{kh} \in C(\bar{I}, H) \,\middle|\, v_{kh}\big|_{I_m} \in X_{k,h}^{r,s,m} \right\} \subset X_k^r.$$

We note that this definition of $X_{k,h}^{r,s,m}$ ensures the continuity of all functions belonging to $X_{k,h}^{r,s}$. This is due to the fact that the vanishing spatial degrees of freedom in $V_h^{s,m-1}$ are coupled only with the temporal basis function τ_0 which vanishes at the right end of each subinterval.

Using these spaces, the cG(s)cG(r) formulation of problem (1.3) reads: Find $u_{kh} \in X_{k,h}^{r,s}$ such that

$$(2.5)\quad (\partial_t u_{kh}, \varphi) + a(u_{kh})(\varphi) + (u_{kh}(0), \varphi_0^-)_H = (f, \varphi) + (u^0, \varphi_0^-)_H \quad \forall \varphi \in \widetilde{X}_{k,h}^{r-1,s}.$$

Remark 2.4. By this definition of the trial and test space for the cG(s)cG(r) discretization, we ensure that not only do both spaces have the same dimension but also that each time step results in a quadratic system of equations. Furthermore, if all spatial meshes and consequently all spatial finite element spaces are equal, this definition of the space $X_{k,h}^{r,s}$ coincides with the familiar one:

$$X_{k,h}^{r,s} = \left\{ v_{kh} \in C(\bar{I}, H) \,\middle|\, v_{kh}\big|_{I_m} \in \mathcal{P}_r(I_m, V_h^s) \right\}.$$

Remark 2.5. In the definition of $X^{r,s}_{k,h}$, we have made no restrictions concerning the spaces $V^{s,m-1}_h$ and $V^{s,m}_h$. Thus, one can for both continuous and discontinuous Galerkin time discretizations with dynamically changing meshes easily modify the definitions of X^r_k, $\widetilde{X}^r_k$, $X^{r,s}_{k,h}$, and $\widetilde{X}^{r,s}_{k,h}$ to allow a different degree of time polynomials on each time interval in the sense of a *hp*-method (in time).

Remark 2.6. Due to the fact that the test space is discontinuous in time for both dG(r)and cG(r) discretizations, these methods (although globally formulated) can be interpreted as time-stepping schemes. The dG(r) discretization in time is known to be strongly A-stable, and the cG(r) method leads to an A-stable scheme. For a priori error estimates for these schemes, we refer, e.g., to [13, 14, 18].

To illustrate the fact that dG(r) and cG(r) lead to time-stepping schemes, we present the time-stepping formulation for the low order methods dG(0) and cG(1). For the cG(s)dG(0) formulation, we define

$$U_0 = u_{kh}(0) \in V^{s,0}_h \quad \text{and} \quad U_m = u_{kh}\big|_{I_m} \in V^{s,m}_h, \quad i = 1, \dots, M,$$

and obtain the following discrete equations:

- $m = 0$:

$$(U_0, \psi)_H = (u^0, \psi)_H \quad \forall \psi \in V^{s,0}_h. \tag{2.6a}$$

- $m = 1, \dots, M$:

$$(U_m, \psi)_H + k_m \bar{a}(U_m)(\psi) = (U_{m-1}, \psi)_H + \int_{I_m} (f(t), \psi)_H \, dt \quad \forall \psi \in V^{s,m}_h. \tag{2.6b}$$

Remark 2.7. This scheme is a variant of the backward Euler scheme. If the time integrals are approximated by the box rule, then the resulting scheme is equivalent to the backward Euler method. However, a better approximation of these time integrals leads to a scheme which allows for better error estimates with respect to the required smoothness of the solution and in the case of long time integration ($T \gg 1$); see, e.g., [13].

For the cG(s)cG(1) formulation, we set

$$U_m = u_{kh}(t_m) \in V^{s,m}_h, \quad m = 0, 1, \dots, M,$$

and obtain:

- $m = 0$:

$$(U_0, \psi)_H = (u^0, \psi)_H \quad \forall \psi \in V^{s,0}_h. \tag{2.7a}$$

- $m = 1, \dots, M$:

$$\begin{aligned} (U_m, \psi)_H + \int_{I_m} \bar{a}\Big(\frac{1}{k_m}((t - t_{m-1})U_m - (t - t_m)U_{m-1})\Big)(\psi)\, dt \\ = (U_{m-1}, \psi)_H + \int_{I_m} (f(t), \psi)_H \, dt \quad \forall \psi \in V^{s,m}_h. \end{aligned} \tag{2.7b}$$

This scheme is a variant of the well-known Crank–Nicolson scheme. If the integrals in (2.7b) are approximated by the trapezoidal rule, one obtains the usual Crank–Nicolson method:

- $m = 0$:

$$(U_0, \psi)_H = (u^0, \psi)_H \quad \forall \psi \in V_h^{s,0}. \tag{2.8a}$$

- $m = 1, \dots, M$:

$$\begin{aligned}(U_m, \psi)_H + \frac{k_m}{2}\bar{a}(U_m)(\psi) + \frac{k_m}{2}\bar{a}(U_{m-1})(\psi) \\ = (U_{m-1}, \psi)_H + \frac{k_m}{2}((f(t_{m-1}), \psi)_H + (f(t_m), \psi)_H) \quad \forall \psi \in V_h^{s,m}.\end{aligned} \tag{2.8b}$$

2.3. Practical aspects of computing on dynamic meshes. In this section, we discuss some algorithmical details of the solution process on dynamic meshes. This is done exemplarily for the cG(s)dG(0) formulation (2.6). The discussion can be easily adapted to the general case. For the computation of U_m in (2.6b), we will discuss problems arising from the term $(U_{m-1}, \psi)_H$. We have to evaluate the inner product of a function in $V_h^{s,m-1}$ with one in $V_h^{s,m}$. Using nodal basis functions in both spaces, this reduces to the problem of evaluating the inner product of basis functions $\psi^{m-1} \in V_h^{s,m-1}$ with basis functions $\psi^m \in V_h^{s,m}$. These integrals cannot be evaluated cellwise by quadrature rules because the integrand (especially ψ^{m-1}) is not necessarily smooth on each cell $K \in \mathcal{T}_h^m$.

To overcome this problem, we require that all meshes $\mathcal{T}_h^m$ result from one original mesh $\widetilde{\mathcal{T}_h}$ by hierarchical refinement or coarsening. Thus, we can build up a temporary mesh $\mathcal{T}_h^{m-1/2}$ as a common refinement of $\mathcal{T}_h^{m-1}$ and $\mathcal{T}_h^m$; see Figure 2.1. Then, looking at the corresponding finite element space $V_h^{s,m-1/2}$, there clearly holds $V_h^{s,m-1} \subset V_h^{s,m-1/2}$ as well as $V_h^{s,m} \subset V_h^{s,m-1/2}$. Therefore, we can express $\psi^{m-1} \in V_h^{s,m-1}$ and $\psi^m \in V_h^{s,m}$ by a linear combination of the nodal basis functions $\psi_i^{m-1/2} \in V_h^{s,m-1/2}$:

$$\psi^{m-1} = \sum_{i=1}^{N} \alpha_i \psi_i^{m-1/2}, \quad \psi^m = \sum_{i=1}^{N} \beta_i \psi_i^{m-1/2},$$

with $\alpha_i, \beta_i \in \mathbb{R}$ and $N = \dim V_h^{s,m-1/2}$. But since $V_h^{s,m-1}$, $V_h^{s,m}$, and $V_h^{s,m-1/2}$ are finite element spaces, the nodal basis functions have only local support, and therefore most of the coefficients α_i and β_i are zero. After these preparations, we can evaluate the integral cellwise on $\mathcal{T}_h^{m-1/2}$ because

$$(\psi^{m-1}, \psi^m)_H = \sum_{i=1}^{N} \alpha_i (\psi_i^{m-1/2}, \psi^m)_H = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \beta_j (\psi_i^{m-1/2}, \psi_j^{m-1/2})_H$$

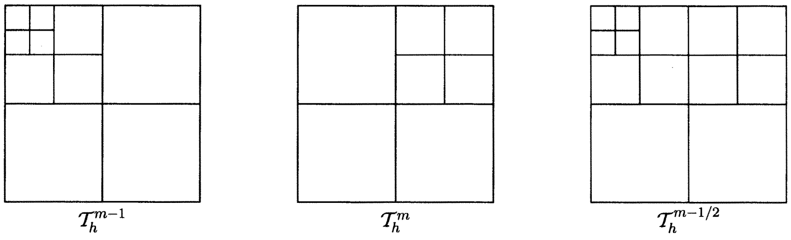


FIG. 2.1. *Construction of temporary mesh* $\mathcal{T}_h^{m-1/2}$.

and $(\psi_i^{m-1/2}, \psi_j^{m-1/2})_H$ can be evaluated cellwise. A similar strategy can be used to evaluate terms such as $\bar{a}(U_{m-1})(\psi^m)$ arising in the cG(s)cG(1) discretization.

3. A posteriori error estimation.

In this section, we derive an a posteriori error estimator for the errors arising due to the discretization in time and space with respect to a functional J. To this end, we first recall an abstract result from [4].

PROPOSITION 3.1. *Let Y be a function space and L a three times differentiable functional on Y. We seek a stationary point y of L on Y, that is,*

$$L'(y)(\delta y) = 0 \quad \forall \delta y \in Y. \tag{3.1}$$

This equation is approximated by a Galerkin method using a finite-dimensional subspace $Y_0 \subset Y$. The discrete problem seeks $y_0 \in Y_0$ satisfying

$$L'(y_0)(\delta y_0) = 0 \quad \forall \delta y_0 \in Y_0.$$

Then we have for arbitrary $\tilde{y}_0 \in Y_0$ the error representation

$$L(y) - L(y_0) = \frac{1}{2} L'(y_0)(y - \tilde{y}_0) + \mathcal{R}, \tag{3.2}$$

where the remainder $\mathcal{R}$ is given by

$$\mathcal{R} = \int_0^1 L'''(y_0 + se)(e, e, e) \cdot s \cdot (s-1)\, ds,$$

with the error $e := y - y_0$.

In what follows, we present the derivation of an a posteriori error estimator in terms of a functional

$$J(u) - J(u_{kh})$$

for the fully discrete problem in the case of a dG time discretization. To this end, we introduce the Lagrangian $\mathcal{L}\colon X \times X \to \mathbb{R}$, defined by

$$\mathcal{L}(u, z) := J(u) + (f - \partial_t u, z) - a(u)(z) - (u(0) - u^0, z(0))_H.$$

The Lagrange functional $\widetilde{\mathcal{L}}\colon \widetilde{X}_k^r \times \widetilde{X}_k^r \to \mathbb{R}$ associated with the dG(r) discretization is defined by

$$\widetilde{\mathcal{L}}(u_k, z_k) := J(u_k) + (f, z_k) - \sum_{m=1}^{M} \int_{I_m} (\partial_t u_k, z_k)_H \, dt - a(u_k)(z_k) \\ - \sum_{m=0}^{M-1} ([u_k]_m, z_{k,m}^+)_H - (u_{k,0}^- - u^0, z_{k,0}^-)_H.$$

To separate the influences of the temporal and spatial discretizations, we split

$$J(u) - J(u_{kh}) = \big(J(u) - J(u_k)\big) + \big(J(u_k) - J(u_{kh})\big),$$

where $u \in X$ is the solution of the continuous problem (1.3), $u_k \in \widetilde{X}_k^r$ is the solution of the time discretized problem (2.1), and $u_{kh} \in \widetilde{X}_{k,h}^{r,s}$ is the solution of the fully discretized problem (2.3).

Theorem 3.2. *Let $(u,z) \in X \times X$, $(u_k, z_k) \in \widetilde{X}_k^r \times \widetilde{X}_k^r$, and $(u_{kh}, z_{kh}) \in \widetilde{X}_{k,h}^{r,s} \times \widetilde{X}_{k,h}^{r,s}$ be stationary points of $\mathcal{L}$ and $\widetilde{\mathcal{L}}$, respectively, i.e.,*

$$\begin{aligned} \mathcal{L}'(u,z)(\delta u, \delta z) = \widetilde{\mathcal{L}}'(u,z)(\delta u, \delta z) = 0 \quad & \forall (\delta u, \delta z) \in X \times X, \\ \widetilde{\mathcal{L}}'(u_k, z_k)(\delta u_k, \delta z_k) = 0 \quad & \forall (\delta u_k, \delta z_k) \in \widetilde{X}_k^r \times \widetilde{X}_k^r, \\ \widetilde{\mathcal{L}}'(u_{kh}, z_{kh})(\delta u_{kh}, \delta z_{kh}) = 0 \quad & \forall (\delta u_{kh}, \delta z_{kh}) \in \widetilde{X}_{k,h}^{r,s} \times \widetilde{X}_{k,h}^{r,s}. \end{aligned}$$

Then we have the following error representations:

$$\begin{aligned} J(u) - J(u_k) &= \frac{1}{2} \widetilde{\mathcal{L}}'(u_k, z_k)(u - \tilde{u}_k, z - \tilde{z}_k) + \mathcal{R}_k, \\ J(u_k) - J(u_{kh}) &= \frac{1}{2} \widetilde{\mathcal{L}}'(u_{kh}, z_{kh})(u_k - \tilde{u}_{kh}, z_k - \tilde{z}_{kh}) + \mathcal{R}_h. \end{aligned}$$

Here $(\tilde{u}_k, \tilde{z}_k) \in \widetilde{X}_k^r \times \widetilde{X}_k^r$ and $(\tilde{u}_{kh}, \tilde{z}_{kh}) \in \widetilde{X}_{k,h}^{r,s} \times \widetilde{X}_{k,h}^{r,s}$ can be chosen arbitrarily, and the remainder terms $\mathcal{R}_k$ and $\mathcal{R}_h$ have the same form as given in Proposition 3.1 *for $L = \widetilde{\mathcal{L}}$.*

Proof. Since $u \in X$, $u_k \in X_k^r$, and $u_{kh} \in \widetilde{X}_{k,h}^{r,s}$ are the solutions of the continuous, time discrete, and fully discrete problems, respectively, we have for arbitrary $z \in X$, $z_k \in \widetilde{X}_k^r$, and $z_{kh} \in \widetilde{X}_{k,h}^{r,s}$:

$$J(u) - J(u_k) = \widetilde{\mathcal{L}}(u,z) - \widetilde{\mathcal{L}}(u_k, z_k), \tag{3.3}$$

$$J(u_k) - J(u_{kh}) = \widetilde{\mathcal{L}}(u_k, z_k) - \widetilde{\mathcal{L}}(u_{kh}, z_{kh}), \tag{3.4}$$

where we have used the identity

$$J(u) = \mathcal{L}(u,z) = \widetilde{\mathcal{L}}(u,z),$$

which follows from the fact that the solution $u \in X$ of the continuous problem is continuous and hence the additional jump terms in $\widetilde{\mathcal{L}}$ compared to $\mathcal{L}$ vanish.

To apply the abstract error representation (3.2), we choose the spaces Y and Y_0 as follows:

$$\begin{aligned} &\text{for (3.3):} \quad Y = (X + \widetilde{X}_k^r) \times (X + \widetilde{X}_k^r), \quad && Y_0 = \widetilde{X}_k^r \times \widetilde{X}_k^r, \\ &\text{for (3.4):} \quad Y = \widetilde{X}_k^r \times \widetilde{X}_k^r, && Y_0 = \widetilde{X}_{k,h}^{r,s} \times \widetilde{X}_{k,h}^{r,s}. \end{aligned}$$

While the choice of the second pairing $Y_0 \subset Y$ is obvious since we have $\widetilde{X}_{k,h}^{r,s} \subset \widetilde{X}_k^r$, for the first pairing we have to take into account that $\widetilde{X}_k^r \not\subset X$. Thus, to fulfill the prerequisites of Proposition 3.1, we have chosen Y in this way. The validity of (3.1) can be shown by a density argument. □

In the above theorem, we require that (u,z), (u_k, z_k), and (u_{kh}, z_{kh}) are stationary points of $\widetilde{\mathcal{L}}$ for continuous, semidiscrete, and discrete test functions, respectively. To this end, the so-called *dual variables* $z \in X$, $z_k \in \widetilde{X}_k^r$, and $z_{kh} \in \widetilde{X}_{k,h}^{r,s}$ have to fulfill the following *dual equations*:

$$(\partial_t \varphi, z) + a'(u)(\varphi, z) + (\varphi(0), z(0)) = J'(u)(\varphi) \quad \forall \varphi \in X, \tag{3.5}$$

$$\begin{aligned} \sum_{m=1}^{M} \int_{I_m} (\partial_t \varphi, z_k)_H \, dt + a'(u_k)(\varphi, z_k) + \sum_{m=0}^{M-1} ([\varphi]_m, z_{k,m}^+)_H + (\varphi_0^-, z_{k,0}^-)_H \\ = J'(u_k)(\varphi) \quad \forall \varphi \in \widetilde{X}_k^r, \end{aligned} \tag{3.6}$$

and

$$\sum_{m=1}^{M}\int_{I_m}(\partial_t\varphi, z_{kh})_H\,dt + a'(u_{kh})(\varphi, z_{kh}) + \sum_{m=0}^{M-1}([\varphi]_m, z_{kh,m}^{+})_H + (\varphi_0^{-}, z_{kh,0}^{-})_H = J'(u_{kh})(\varphi) \quad \forall\varphi\in\widetilde{X}_{k,h}^{r,s}. \tag{3.7}$$

The time-stepping formulation of (3.7) in the cG(s)dG(0) case can be directly obtained; see [2]. To this end, we use the precise form of J (1.4) and set

$$Z_0 = z_{kh}(0)\in V_h^{s,0} \quad\text{and}\quad Z_m = z_{kh}\big|_{I_m}\in V_h^{s,m},\quad i=1,\dots,M,$$

and obtain the following discrete equations:

- $m=M$:
(3.8a)
$$(\psi, Z_M)_H + k_M\bar{a}'(U_M)(\psi, Z_M) = k_M J_1'(U_M)(\psi) + J_2'(U_M)(\psi) \quad \forall\psi\in V_h^{s,M}.$$
- $m=M-1,\dots,1$:
(3.8b)
$$(\psi, Z_m)_H + k_m\bar{a}'(U_m)(\psi, Z_m) = (\psi, Z_{m+1})_H + k_m J_1'(U_m)(\psi) \quad \forall\psi\in V_h^{s,m}.$$
- $m=0$:
$$(\psi, Z_0)_H = (\psi, Z_1)_H \quad \forall\psi\in V_h^{s,0}. \tag{3.8c}$$

Remark 3.1. To solve the above equations for determining the dual solution z_{kh}, the (already computed) primal solution u_{kh} has to be stored in all time intervals. For large problems the required storage can exceed the size of the main storage (RAM). There are in general two possibilities to cope with this problem: One can use storage reduction techniques (checkpointing) allowing one to reduce the required storage to $\mathcal{O}(\log(M))$. However, such techniques lead to additional computations. Another possibility is to store the solution u_{kh} on a hard disc. The additional time required for the communication with the hard disc is usually negligible in comparison with the CPU time required for the computation of the dual solution.

Introducing the primal and dual residuals

$$\begin{aligned}\tilde{\rho}(u)(\varphi) &:= \widetilde{\mathcal{L}}_z'(u,z)(\varphi),\\ \tilde{\rho}^*(u,z)(\varphi) &:= \widetilde{\mathcal{L}}_u'(u,z)(\varphi),\end{aligned}$$

the result of Theorem 3.2 can be formulated as

$$\begin{aligned}J(u)-J(u_k) &\approx \frac{1}{2}\Big(\tilde{\rho}(u_{kh})(z-\tilde{z}_k) + \tilde{\rho}^*(u_{kh}, z_{kh})(u-\tilde{u}_k)\Big),\\ J(u_k)-J(u_{kh}) &\approx \frac{1}{2}\Big(\tilde{\rho}(u_{kh})(z_k-\tilde{z}_{kh}) + \tilde{\rho}^*(u_{kh}, z_{kh})(u_k-\tilde{u}_{kh})\Big),\end{aligned}$$

where we have replaced $\tilde{\rho}(u_k)(z-\tilde{z}_k)$ by $\tilde{\rho}(u_{kh})(z-\tilde{z}_k)$ as well as $\tilde{\rho}^*(u_k,z_k)(u-\tilde{u}_k)$ by $\tilde{\rho}^*(u_{kh},z_{kh})(u-\tilde{u}_k)$ and have omitted the remainder term $\mathcal{R}$.

Remark 3.2. The above replacement of $\tilde{\rho}(u_k)(z-\tilde{z}_k)$ by $\tilde{\rho}(u_{kh})(z-\tilde{z}_k)$ is necessary for the evaluation of this term. However, this procedure may introduce an additional error of the form

$$\widetilde{\mathcal{L}}'(x_k)(x-\tilde{x}_k) - \widetilde{\mathcal{L}}'(x_{kh})(x-\tilde{x}_k) = \widetilde{\mathcal{L}}''(\xi)(x-\tilde{x}_k, x_k - x_{kh})$$

with some $\xi \in [x_k, x_{kh}]$. This error is usually of "higher order." For example, for the cG(1)dG(0) discretization of the heat equation and choice of $\tilde{x}_k$ to be some interpolant of x, this error can be shown to be of order $\mathcal{O}(kh^2)$, whereas the error $J(u) - J(u_{kh})$ is not better than $\mathcal{O}(k + h^2)$.

In the case of the cG(s)cG(r) discretization, one obtains similar error representations:

$$\begin{aligned} J(u) - J(u_k) &= \frac{1}{2}\mathcal{L}'(u_k, z_k)(u - \tilde{u}_k, z - \tilde{z}_k) + \mathcal{R}_k, \\ J(u_k) - J(u_{kh}) &= \frac{1}{2}\mathcal{L}'(u_{kh}, z_{kh})(u_k - \tilde{u}_{kh}, z_k - \tilde{z}_{kh}) + \mathcal{R}_h. \end{aligned}$$

For the time-stepping formulation of the corresponding dual equation, we refer to [2].

4. Numerical realization.

4.1. Evaluation of the error estimator. In this subsection, we describe the numerical evaluation of the a posteriori error estimator derived in the last section for the cG(1)cG(1) and cG(1)dG(0) discretizations. This means that we consider functions which are piecewise constant or linear in time and bi/trilinear in space. Again, we present the concrete terms only for the dG(0) time discretization, and the cG(1) time discretization can be treated similarly.

The error estimates in the previous section involve interpolation errors of time ($\tilde{v}_k := i_k v$) and space ($\tilde{v}_{kh} := i_h v_k$). We approximate these terms using interpolations into higher order finite element spaces. To this end, we introduce linear operators Π_k and Π_h which map the computed solutions to approximations of the interpolation errors:

$$\begin{aligned} z - \tilde{z}_k &\approx \Pi_k z_k, \qquad & u - \tilde{u}_k &\approx \Pi_k u_k, \\ z_k - \tilde{z}_{kh} &\approx \Pi_h z_{kh}, & u_k - \tilde{u}_{kh} &\approx \Pi_h u_{kh}. \end{aligned}$$

In the considered cases of a cG(1)dG(0) or cG(1)cG(1) discretization, the linear operators Π_k and Π_h are chosen as

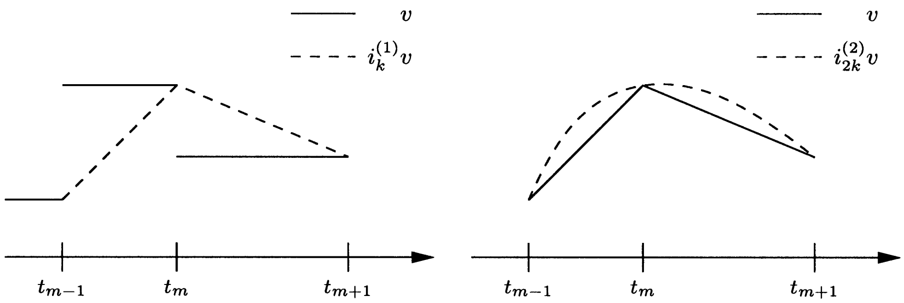
$$\Pi_k := i_k^{(1)} - \mathrm{id}, \text{ with } i_k^{(1)} : \widetilde{X}_k^0 \to X_k^1, \quad \Pi_h := i_{2h}^{(2)} - \mathrm{id}, \text{ with } i_{2h}^{(2)} : \widetilde{X}_{k,h}^{0,1} \to \widetilde{X}_{k,2h}^{0,2}$$

for the cG(1)dG(0) discretization and as

$$\Pi_k := i_{2k}^{(2)} - \mathrm{id}, \text{ with } i_{2k}^{(2)} : X_k^1 \to X_{2k}^2, \quad \Pi_h := i_{2h}^{(2)} - \mathrm{id}, \text{ with } i_{2h}^{(2)} : X_{k,h}^{1,1} \to X_{k,2h}^{1,2}$$

for the cG(1)cG(1) discretization, respectively. Here the piecewise linear and piecewise quadratic interpolation operators $i_k^{(1)}$ and $i_{2k}^{(2)}$ are defined through

$$i_k^{(1)} v(t) := \frac{t_m - t}{k_m} v(t_{m-1}) + \frac{t - t_{m-1}}{k_m} v(t_m) \quad \text{for } t \in (t_{m-1}, t_m]$$



(a) Piecewise linear interpolation (b) Piecewise quadratic interpolation

FIG. 4.1. *Interpolation operators.*

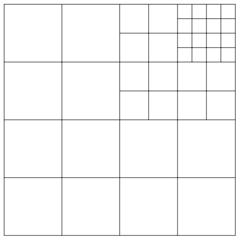


FIG. 4.2. *Mesh with patch structure.*

and

$$i_{2k}^{(2)}v(t) := \frac{(t_m - t)(t_{m+1} - t)}{k_m(k_m + k_{m+1})}v(t_{m-1}) + \frac{(t - t_{m-1})(t_{m+1} - t)}{k_m k_{m+1}}v(t_m) + \frac{(t - t_{m-1})(t - t_m)}{(k_m + k_{m+1})k_m}v(t_{m+1}) \quad \text{for } t \in (t_{m-1}, t_{m+1}];$$

see also Figure 4.1. The piecewise bi/triquadratic spatial interpolation $i_{2h}^{(2)}$ can easily be computed if the spatial mesh provides a patch structure; i.e., one can always combine four/eight adjacent cells to one macrocell on which the bi/triquadratic interpolation can be defined. An example of such a mesh is depicted in Figure 4.2.

Finally, we obtain the following computable a posteriori error estimator:

$$J(u) - J(u_{kh}) \approx \eta_k + \eta_h,$$

with

$$\eta_k := \frac{1}{2}\Big(\tilde{\rho}(u_{kh})(\Pi_k z_{kh}) + \tilde{\rho}^*(u_{kh}, z_{kh})(\Pi_k u_{kh})\Big),$$

$$\eta_h := \frac{1}{2}\Big(\tilde{\rho}(u_{kh})(\Pi_h z_{kh}) + \tilde{\rho}^*(u_{kh}, z_{kh})(\Pi_h u_{kh})\Big).$$

To give an impression of the terms which have to be evaluated for the presented error estimators, we present for the cG(1)dG(0) discretization the explicit form of the primal and dual residuals. With the notation $U_0 := u_{kh}(0)$, $U_m := u_{kh}\big|_{I_m}$, and

$Z_0 := z_{kh}(0)$, $Z_m := z_{kh}\big|_{I_m}$, we obtain

$$\begin{aligned}
\tilde\rho(u_{kh})(\Pi_k z_{kh}) &= \sum_{m=1}^{M}\Bigg\{(U_m - U_{m-1}, Z_m - Z_{m-1})_H + \frac{k_m}{2}\bar a(U_m)(Z_m - Z_{m-1}) \\
&\qquad + \int_{I_m}\frac{t - t_m}{k_m}(f, Z_m - Z_{m-1})_H\,dt\Bigg\}, \\
\tilde\rho^*(u_{kh}, z_{kh})(\Pi_k u_{kh}) &= \sum_{m=1}^{M}\Bigg\{\frac{k_m}{2}\bar a'(U_m)(U_m - U_{m-1}, Z_m) \\
&\qquad - \frac{k_m}{2}J_1'(U_m)(U_m - U_{m-1})\Bigg\}, \\
\tilde\rho(u_{kh})(\Pi_h z_{kh}) &= \sum_{m=1}^{M}\Bigg\{\int_{I_m}(f, i_{2h}^{(2)}Z_m - Z_m)_H\,dt - k_m\bar a(U_m)(i_{2h}^{(2)}Z_m - Z_m) \\
&\qquad - (U_m - U_{m-1}, i_{2h}^{(2)}Z_m - Z_m)_H\Bigg\} - (U_0 - u^0, i_{2h}^{(2)}Z_0 - Z_0)_H, \\
\tilde\rho^*(u_{kh}, z_{kh})(\Pi_h u_{kh}) &= \sum_{m=1}^{M}\Big\{k_m J_1'(U_m)(i_{2h}^{(2)}U_m - U_m) - k_m\bar a'(U_m)(i_{2h}^{(2)}U_m - U_m, Z_m) \\
&\qquad + (i_{2h}^{(2)}U_{m-1} - U_{m-1}, Z_m - Z_{m-1})_H\Big\} \\
&\qquad + J_2'(U_M)(i_{2h}^{(2)}U_M - U_M) - (i_{2h}^{(2)}U_M - U_M, Z_M)_H.
\end{aligned}$$

Remark 4.1. The terms that have to be evaluated for the cG(1)cG(1) discretization are very similar, and the evaluation can be treated as presented here for the cG(1)dG(0) discretization. In practice, however, one often uses the Crank–Nicolson scheme (2.8), which is easier to implement. For precise error estimation in this situation, one has to take into account the error due to the numerical integration.

The continuous solution $x = (u, z)$ is a stationary point of the Lagrangian $\mathcal{L}$, whereas the semidiscrete solution $x_k = (u_k, z_k)$ is a stationary point of a modified Lagrangian $\mathcal{L}_k$ obtained by applying the trapezoidal rule for the time integrals in $\mathcal{L}$. One obtains:

$$J(u) - J(u_k) = \mathcal{L}(x) - \mathcal{L}_k(x_k) = \big(\mathcal{L}(x) - \mathcal{L}(x_k)\big) + \big(\mathcal{L}(x_k) - \mathcal{L}_k(x_k)\big).$$

The first term is estimated using the technique presented above, and the second term can be computed using a better numerical quadrature for the approximation of $\mathcal{L}(x_k)$. We emphasize that we use a higher order numerical quadrature (two-point Gauss quadrature in the numerical examples below) only for the evaluation of the error estimator. The solution process is based on the trapezoidal rule leading to the usual Crank–Nicolson formulation which saves an amount of numerical effort and provides the same order of accuracy compared to the cG(1) discretization.

The presented a posteriori error estimators are used for two purposes: First, we gain access to the discretization error, and, second, we improve the accuracy by local refinement of the spatial and temporal meshes. For the second aim, we have to localize the information provided by the error estimator to cellwise or nodewise contributions (local error indicators). To this end, we split up the error estimates η_k and η_h into their contribution on each subinterval I_m by

$$\eta_k = \sum_{m=1}^{M} \eta_k^m \quad \text{and} \quad \eta_h = \sum_{m=0}^{M} \eta_h^m,$$

where

$$\begin{aligned}\eta_k^m := \frac{1}{2}\Bigg\{ & (U_m - U_{m-1}, Z_m - Z_{m-1})_H + \frac{k_m}{2}\bar{a}(U_m)(Z_m - Z_{m-1}) \\ & + \int_{I_m} \frac{t - t_m}{k_m}(f, Z_m - Z_{m-1})_H \, dt + \frac{k_m}{2}\bar{a}'(U_m)(U_m - U_{m-1}, Z_m) \\ & - \frac{k_m}{2} J_1'(U_m)(U_m - U_{m-1}) \Bigg\}, \quad m = 1, \dots, M,\end{aligned}$$

and

$$\begin{aligned}\eta_h^0 &:= \frac{1}{2}\Big\{ (i_{2h}^{(2)} U_0 - U_0, Z_1 - Z_0)_H - (U_0 - u^0, i_{2h}^{(2)} Z_0 - Z_0)_H \Big\}, \\ \eta_h^m &:= \frac{1}{2}\Bigg\{ \int_{I_m} (f, i_{2h}^{(2)} Z_m - Z_m)_H \, dt - k_m \bar{a}(U_m)(i_{2h}^{(2)} Z_m - Z_m) \\ &\quad - (U_m - U_{m-1}, i_{2h}^{(2)} Z_m - Z_m)_H + k_m J_1'(U_m)(i_{2h}^{(2)} U_m - U_m) \\ &\quad - k_m \bar{a}'(U_m)(i_{2h}^{(2)} U_m - U_m, Z_m) \\ &\quad + (i_{2h}^{(2)} U_m - U_m, Z_{m+1} - Z_m)_H \Bigg\}, \quad m = 1, \dots, M-1, \\ \eta_h^M &:= \frac{1}{2}\Bigg\{ \int_{I_M} (f, i_{2h}^{(2)} Z_M - Z_M)_H \, dt - k_M \bar{a}(U_M)(i_{2h}^{(2)} Z_M - Z_M) \\ &\quad - (U_M - U_{M-1}, i_{2h}^{(2)} Z_M - Z_M)_H + k_M J_1'(U_M)(i_{2h}^{(2)} U_M - U_M) \\ &\quad - k_M \bar{a}'(U_M)(i_{2h}^{(2)} U_M - U_M, Z_M) + J_2'(U_M)(i_{2h}^{(2)} U_M - U_M) \\ &\quad - (i_{2h}^{(2)} U_M - U_M, Z_M)_H \Bigg\}.\end{aligned}$$

The error estimators η_h^m have to be localized to cellwise contributions in order to be used as error indicators within an adaptive algorithm. To this end, we proceed as follows: We first consider the nodal basis $\{\varphi_i^m\}$ of $V_h^{1,m}$ and quadratic functions $\psi_i^m := i_{2h}^{(2)} \varphi_i^m \in V_{2h}^{2,m}$. Using the representations

$$u_{kh}\big|_{I_m} = \sum_{i=1}^{N_m} \varphi_i^m U_i^m, \quad z_{kh}\big|_{I_m} = \sum_{i=1}^{N_m} \varphi_i^m Z_i^m$$

with the nodal vectors $U^m, Z^m \in \mathbb{R}^{N_m}$, we can rewrite the error estimator as

$$\eta_h^m = \frac{1}{2}\{\langle \Psi_m, Z^m \rangle + \langle \Psi_m^*, U^m \rangle\},$$

where $\langle\cdot,\cdot\rangle$ denotes the Euclidian inner product on $\mathbb{R}^{N_m}$ and

$$\begin{aligned}\Psi_{m,i} &:= \tilde\rho(u_{kh})(\psi_i^m - \varphi_i^m),\\ \Psi^*_{m,i} &:= \tilde\rho^*(u_{kh}, z_{kh})(\psi_i^m - \varphi_i^m).\end{aligned}$$

These terms are first localized to nodewise contributions. However, a direct localization of this representation would lead to an overestimation of the error due to oscillatory behavior of the residuals. One way to overcome this difficulty is to use integration by parts requiring the evaluation of the strong formulation of the differential equation as well as jump terms of the discrete solution over edges or faces of cells; see, e.g., [3]. Another possibility which we used for numerical examples in section 5 is to introduce a filtering operator $\pi = \mathrm{id} - i_{2h}^{(1)}$, where $i_{2h}^{(1)}\colon \widetilde{X}_{k,h}^{0,1} \to \widetilde{X}_{k,2h}^{0,1}$ is an interpolation operator in the space of bi/trilinear elements on patches. We use the representations of πu_{kh} and πz_{kh} in the basis $\{\varphi_i^m\}$

$$\pi u_{kh}\big|_{I_m} = \sum_{i=1}^{N_m} \varphi_i^m U_i^{\pi,m}, \quad \pi z_{kh}\big|_{I_m} = \sum_{i=1}^{N_m} \varphi_i^m Z_i^{\pi,m}$$

and conclude by employing the Galerkin orthogonality relation:

$$\eta_h^m = \frac{1}{2}\{\langle\Psi_m, Z^{\pi,m}\rangle + \langle\Psi^*_m, U^{\pi,m}\rangle\}.$$

A further localization provides us with

$$\eta_{h,i}^m = \frac{1}{2}\{\Psi_{m,i}, Z_i^{\pi,m} + \Psi^*_{m,i} U_i^{\pi,m}\}, \quad i = 1, \dots, N_m.$$

While η_h is almost independent of the time step size (see Table 5.3 in the next section), η_h^m depends linearly on k_m. To get rid of this dependence, we define a reference time step

$$\hat{k} = \frac{T}{M}$$

and introduce scaled local error indicators by

$$\tilde\eta_{h,i}^m := \frac{\hat{k}}{k_m}\eta_{h,i}^m.$$

For mesh refinement these nodewise contributions are assembled to the corresponding cellwise contributions $\tilde\eta_{h_K}^m$. Finally, we end up with two sets of error indicators, one for the temporal and one for the spatial discretization error:

$$\{\,\eta_k^m \,|\, m = 1, \dots, M\,\} \quad \text{and} \quad \bigcup_{m=0}^{M} \{\,\tilde\eta_{h_K}^m \,\big|\, K \in \mathcal{T}_h^m\,\}.$$

These error indicators are used for the decision which of the cells and of the time intervals have to be refined within the adaptive algorithm described below. For the time refinement, the time intervals are ordered with respect to the absolute values of the indicators:

$$|\eta_k^{m_1}| \ge |\eta_k^{m_2}| \ge \cdots.$$

Then the intervals $m_1, m_2, \dots, m_s$ are bisected where the number s is chosen using one of the known strategies; see, e.g., [4]. The same procedure is used for the refinement of the spatial meshes. We emphasize that we do not adapt each mesh separately but choose cells for refinement from the set of all cells in all meshes $\mathcal{T}_h^0, \mathcal{T}_h^1, \dots, \mathcal{T}_h^M$.

4.2. Adaptive algorithm. The goal of the adaptation of the spatial and temporal discretization has to be the equilibrated reduction of the corresponding discretization errors. If a tolerance (TOL) is given, this can be done by refining both temporal and spatial discretizations as long as the corresponding error estimator is larger than $\frac{\text{TOL}}{2}$. In what follows, we will present an adaptive algorithm which balances the spatial and temporal discretization error and keeps it balanced under refinement of the discretizations without the need of a given TOL. We stop the calculation if the discretization has reached the maximum number of allowed degrees of freedom $K_{\max}$.

The aim of the adaptive algorithm is to obtain a discretization such that

$$|\eta_h| \approx |\eta_k|$$

and to keep this property during further refinement. To this end, we introduce an equilibration factor $c \in (1, 5)$ and propose Algorithm 4.1 presented below. In every cycle of the algorithm, we select cells and time steps for refinement to be adapted by means of local error indicators as described above.

Algorithm 4.1. Adaptive refinement algorithm.

```
Choose an initial space-time discretization $\mathcal{T}_{k_0,h_0}$ and set $n = 0$.
while $\sum_{m=0}^{M_n} N_m \le K_{\max}$ do
  Compute the primal and dual solution $(u_{k_n h_n}, z_{k_n h_n})$.
  Evaluate the a posteriori error estimators $\eta_{k_n}$ and $\eta_{h_n}$.
  if $|\eta_{k_n}| > c|\eta_{h_n}|$ then
    Refine temporal discretization.
  else if $|\eta_{h_n}| > c|\eta_{k_n}|$ then
    Refine spatial discretization.
  else $\{\frac{1}{c} \le \frac{|\eta_{k_n}|}{|\eta_{h_n}|} \le c\}$
    Refine spatial and temporal discretization.
  end if
  Increase $n$.
end while
```

Remark 4.2. We have tested the presented algorithm for different values of the equilibration factor c. Choosing it too small results in a slower reduction of the discretization error, because then often only one discretization is refined although the discretization errors are of similar size. On the other hand, a too large choice of this factor leads to inefficiency, because, for example, the spatial discretization is refined although the temporal discretization error dominates the overall error. For all results presented below, the equilibration factor was chosen $c = 5$.

5. Numerical examples.

In this section, the adaptive algorithm presented above is applied to two numerical examples.

5.1. Example 1.

The configuration for the first example is taken from [15]. It describes the major part of combustion under the low Mach number hypothesis. Under this assumption, the motion of the fluid becomes independent from temperature and species concentration. Hence, one can solve the temperature and the species equation alone by specifying any solenoidal velocity field v which influences these quantities only via a convection term. Especially, $v = 0$ is an important case.

Introducing the dimensionless temperature θ, denoting by Y the species concentration, and assuming constant diffusion coefficients yields

$$\begin{aligned} \partial_t\theta - \Delta\theta &= \omega(\theta, Y) \qquad \text{in } \Omega \times (0, 60), \\ \partial_t Y - \frac{1}{\mathrm{Le}}\Delta Y &= -\omega(\theta, Y) \qquad \text{in } \Omega \times (0, 60), \end{aligned} \tag{5.1}$$

where the Lewis number Le is the ratio of diffusivity of heat and diffusivity of mass. We use a one-species rection mechanism governed by an Arrhenius law

$$\omega(\theta, Y) = \frac{\beta^2}{2\mathrm{Le}} Y \mathrm{e}^{\frac{\beta(\theta-1)}{1+\alpha(\theta-1)}} \tag{5.2}$$

in which an approximation for large activation energy has been employed. The parameter α determines the gas expansion in flows with nonconstant density, whereas the parameter β describes the nondimensional activation energy. This results in a system of two (nonlinear) coupled parabolic equations.

We consider a freely propagating laminar flame described by (5.1) and its response to a heat-absorbing obstacle, a set of cooled parallel rods with a rectangular cross section (see Figure 5.1). The computational domain has width $H = 16$ and length $L = 60$. The obstacle covers half of the width and has length $\frac{L}{4}$. The absorption of heat is modeled by Robin boundary conditions with the heat loss parameter k. On the left boundary of the domain, Dirichlet boundary conditions corresponding to the burnt state are prescribed, while the remaining boundary conditions are of the homogeneous Neumann type. Thus, the boundary conditions are chosen as

$$\begin{aligned} \theta &= 1 && \text{on } \Gamma_\mathrm{D} \times (0, 60), \\ Y &= 0 && \text{on } \Gamma_\mathrm{D} \times (0, 60), \\ \partial_n\theta &= 0 && \text{on } \Gamma_\mathrm{N} \times (0, 60), \\ \partial_n Y &= 0 && \text{on } \Gamma_\mathrm{N} \times (0, 60), \\ \partial_n\theta &= -k\theta && \text{on } \Gamma_\mathrm{R} \times (0, 60), \\ \partial_n Y &= 0 && \text{on } \Gamma_\mathrm{R} \times (0, 60). \end{aligned}$$

The initial condition is the analytical solution of a one-dimensional right-traveling flame in the limit $\beta \to \infty$ located left of the obstacle:

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

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

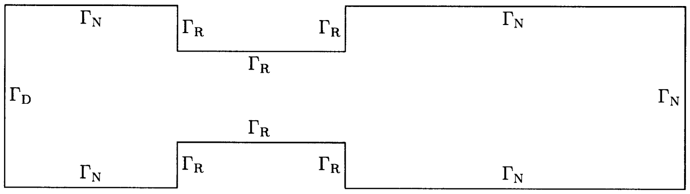


FIG. 5.1. *Example 1: Computational domain Ω.*

For the computations, the occurring parameters are set to

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

To use the same notations as in the previous sections of this article, we define the pair of solution components $u:=(\theta,Y)\in\hat{u}+X^2$, where the space X is defined using the spaces V and H as given by (1.1). The function $\hat{u}$ is defined to fulfill the prescribed Dirichlet boundary conditions as $\hat{u}\big|_{\Gamma_\mathrm{D}}=(1,0)$.

We consider the functional

$$J(u)=\frac{1}{60|\Omega|}\int_0^{60}\int_\Omega \omega(\theta,Y)\,dx\,dt$$

describing the mean reaction rate (5.2) and focus on reducing the error $J(u)-J(u_{kh})$. The corresponding dual equation for $z=(z^\theta,z^Y)$ has the following form:

$$\begin{aligned} -\partial_t z^\theta-\Delta z^\theta &= \left(\frac{1}{60|\Omega|}+z^\theta-z^Y\right)\omega'_\theta(\theta,Y) && \text{in } \Omega\times(0,60),\\ -\partial_t z^Y-\frac{1}{\mathrm{Le}}\Delta z^Y &= \left(\frac{1}{60|\Omega|}+z^\theta-z^Y\right)\omega'_Y(\theta,Y) && \text{in } \Omega\times(0,60), \end{aligned} \tag{5.3}$$

where

$$\omega'_\theta(\theta,Y)=\frac{\beta^2}{2\mathrm{Le}}\frac{\beta}{\big(1+\alpha(\theta-1)\big)^2}Y\mathrm{e}^{\frac{\beta(\theta-1)}{1+\alpha(\theta-1)}} \quad\text{and}\quad \omega'_Y(\theta,Y)=\frac{\beta^2}{2\mathrm{Le}}\mathrm{e}^{\frac{\beta(\theta-1)}{1+\alpha(\theta-1)}}$$

denote the derivatives of $\omega(\theta,Y)$ with respect to θ and Y, respectively. The "initial" and boundary conditions for the dual equations are

$$z^\theta(60,x)=z^Y(60,x)=0 \quad \text{in } \Omega$$

and

$$\begin{aligned} z^\theta &= 0 && \text{on } \Gamma_\mathrm{D}\times(0,60),\\ z^Y &= 0 && \text{on } \Gamma_\mathrm{D}\times(0,60),\\ \partial_n z^\theta &= 0 && \text{on } \Gamma_\mathrm{N}\times(0,60),\\ \partial_n z^Y &= 0 && \text{on } \Gamma_\mathrm{N}\times(0,60),\\ \partial_n z^\theta &= -kz^\theta && \text{on } \Gamma_\mathrm{R}\times(0,60),\\ \partial_n z^Y &= 0 && \text{on } \Gamma_\mathrm{R}\times(0,60). \end{aligned}$$

First, we present the results of computation using the cG(1)dG(0) method. In Table 5.1, we show the development of the discretization error and the a posteriori error estimators during an adaptive run with local refinement in both types of discretizations. Here M denotes the number of time steps, and $N_{\max}:=\max N_m$ denotes the maximum number of nodes in the spatial meshes. The effectivity index given in the last column of this table is defined as usual by

$$I_{\mathrm{eff}}:=\frac{J(u)-J(u_{kh})}{\eta_k+\eta_h}.$$

Table 5.1
Example 1: Local refinement with equilibration for cG(1)dG(0) discretization.

M	$N_{\max}$	η_k	η_h	$\eta_k + \eta_h$	$J(u) - J(u_{kh})$	I_{eff}
256	985	$-7.8 \cdot 10^{-04}$	$4.1 \cdot 10^{-05}$	$-7.439 \cdot 10^{-04}$	$-1.703 \cdot 10^{-03}$	2.2893
396	985	$-7.4 \cdot 10^{-04}$	$2.0 \cdot 10^{-04}$	$-5.394 \cdot 10^{-04}$	$-1.413 \cdot 10^{-03}$	2.6201
616	1427	$-2.5 \cdot 10^{-04}$	$-3.3 \cdot 10^{-04}$	$-5.782 \cdot 10^{-04}$	$-8.718 \cdot 10^{-04}$	1.5076
872	2309	$-1.0 \cdot 10^{-04}$	$-1.4 \cdot 10^{-04}$	$-2.443 \cdot 10^{-04}$	$-4.013 \cdot 10^{-04}$	1.6426
1370	3927	$-5.0 \cdot 10^{-05}$	$-6.8 \cdot 10^{-05}$	$-1.180 \cdot 10^{-04}$	$-1.569 \cdot 10^{-04}$	1.3300
1528	6927	$-4.6 \cdot 10^{-05}$	$-2.8 \cdot 10^{-05}$	$-7.356 \cdot 10^{-05}$	$-8.630 \cdot 10^{-05}$	1.1731
1772	14683	$-4.0 \cdot 10^{-05}$	$-1.1 \cdot 10^{-05}$	$-5.154 \cdot 10^{-05}$	$-5.941 \cdot 10^{-05}$	1.1527

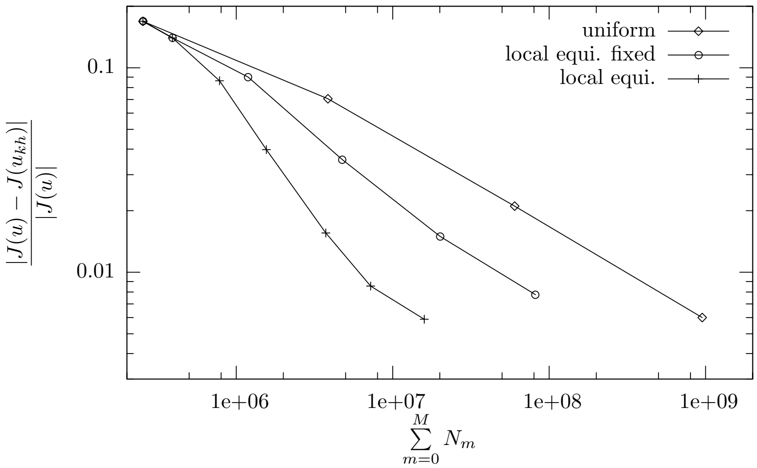


Fig. 5.2. *Example 1: Comparison of different refinement strategies for cG(1)dG(0) discretization.*

It shows that the proposed error estimator provides quantitative information about the discretization error. Table 5.1 also demonstrates the desired equilibration of the temporal and spatial discretization errors.

A comparison of the relative error $\frac{|J(u)-J(u_{kh})|}{|J(u)|}$ for the different refinement strategies is depicted in Figure 5.2:

- "uniform": Here we apply uniform refinement of both temporal and spatial discretizations after each run.
- "local equilibration fixed": Here we combine local refinement of both temporal and spatial discretizations with the proposed equilibration strategy but restrict ourselves to one (locally refined) spatial mesh for the whole time interval.
- "local equilibration": Here we additionally allow the spatial meshes to change in time.

It shows that the uniform refinement needs about 12 times the number of degrees of freedom the local refinement needs to reach a relative error of $6 \cdot 10^{-3}$ and even about 60 times the number of degrees of freedom the local refinement with spatial meshes changing in time.

In the following, we present the results for the cG(1)cG(1) discretization. As in the cG(1)dG(0) case, Table 5.2 shows the development of the discretization error and the a posteriori error estimators during an adaptive run with local refinement in both

TABLE 5.2
Example 1: Local refinement with equilibration for cG(1)cG(1) discretization.

M	$N_{\max}$	η_k	η_h	$\eta_k + \eta_h$	$J(u) - J(u_{kh})$	I_{eff}
256	985	$-6.3{\cdot}10^{-04}$	$1.7{\cdot}10^{-04}$	$-4.532 \cdot 10^{-04}$	$-1.242 \cdot 10^{-03}$	2.7392
420	1423	$-1.3{\cdot}10^{-04}$	$-2.4{\cdot}10^{-04}$	$-3.675 \cdot 10^{-04}$	$-6.757 \cdot 10^{-04}$	1.8388
654	2307	$-2.5{\cdot}10^{-05}$	$-1.5{\cdot}10^{-04}$	$-1.713 \cdot 10^{-04}$	$-2.567 \cdot 10^{-04}$	1.4986
654	3959	$-2.0{\cdot}10^{-05}$	$-5.5{\cdot}10^{-05}$	$-7.504 \cdot 10^{-05}$	$-9.737 \cdot 10^{-05}$	1.2975
706	7701	$-1.5{\cdot}10^{-05}$	$-2.3{\cdot}10^{-05}$	$-3.784 \cdot 10^{-05}$	$-3.782 \cdot 10^{-05}$	0.9993

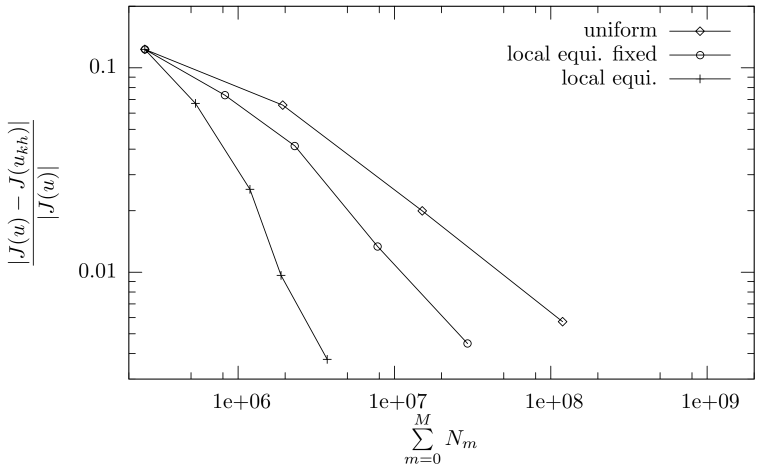


FIG. 5.3. *Example 1: Comparison of different refinement strategies for cG(1)cG(1) discretization.*

temporal and spatial discretizations. The variables M, $N_{\max}$ and I_{eff} are defined as before.

The table again demonstrates the desired equilibration of the different discretization errors and the sufficient quality of the error estimators.

In Figure 5.3, we show a comparison of the relative error $\frac{|J(u)-J(u_{kh})|}{|J(u)|}$ for the three types of refinement strategies, as described before. Compared to Figure 5.2, we see the reduction in needed degrees of freedom to reach the same accuracy as in the cG(1)dG(0) case due to the higher order of the temporal discretization. Moreover, we again observe the advantage of dynamic meshes.

In Table 5.3, we present the numerical justification for splitting the total discretization error into the temporal and spatial parts: The table demonstrates the independence of each part of the error estimator on the refinement of the other part. This feature is especially important to reach an equilibration of the discretization errors by applying the adaptive refinement algorithm described above.

Finally, we present in Figures 5.4, 5.5, and 5.6 for the cG(1)dG(0) discretization the reaction rate ω at different times, the corresponding spatial meshes on which the solution was computed, and the size of the time steps, respectively. Figure 5.5 shows that the meshes are strongly refined in the region where the reactive front is located. Furthermore, we observe that the meshes at the beginning of the time interval are more strongly refined than those at the end. This is due to the fact that errors occurring at the end of the time interval have less influence on the overall discretization error than those occurring at the beginning because of the error accumulation.

TABLE 5.3
Example 1: Independence of one part of the error estimator on the refinement of the other part.

M	N	η_k		η_h	
		dG(0)	cG(1)	dG(0)	cG(1)
512	3761	—	—	$-3.7364\cdot 10^{-04}$	$-3.5395\cdot 10^{-04}$
1024	3761			$-3.6614\cdot 10^{-04}$	$-3.5514\cdot 10^{-04}$
2048	3761			$-3.6459\cdot 10^{-04}$	$-3.5557\cdot 10^{-04}$
4096	3761			$-3.6028\cdot 10^{-04}$	$-3.5567\cdot 10^{-04}$
8192	3761			$-3.5783\cdot 10^{-04}$	$-3.5570\cdot 10^{-04}$
1024	985	$-1.9694\cdot 10^{-04}$	$-4.5279\cdot 10^{-05}$	—	—
1024	3761	$-1.0491\cdot 10^{-04}$	$-2.8233\cdot 10^{-05}$		
1024	14689	$-9.6992\cdot 10^{-05}$	$-2.3308\cdot 10^{-05}$		
1024	58049	$-9.5939\cdot 10^{-05}$	$-2.1107\cdot 10^{-05}$		
1024	230785	$-9.5725\cdot 10^{-05}$	$-2.1039\cdot 10^{-05}$		

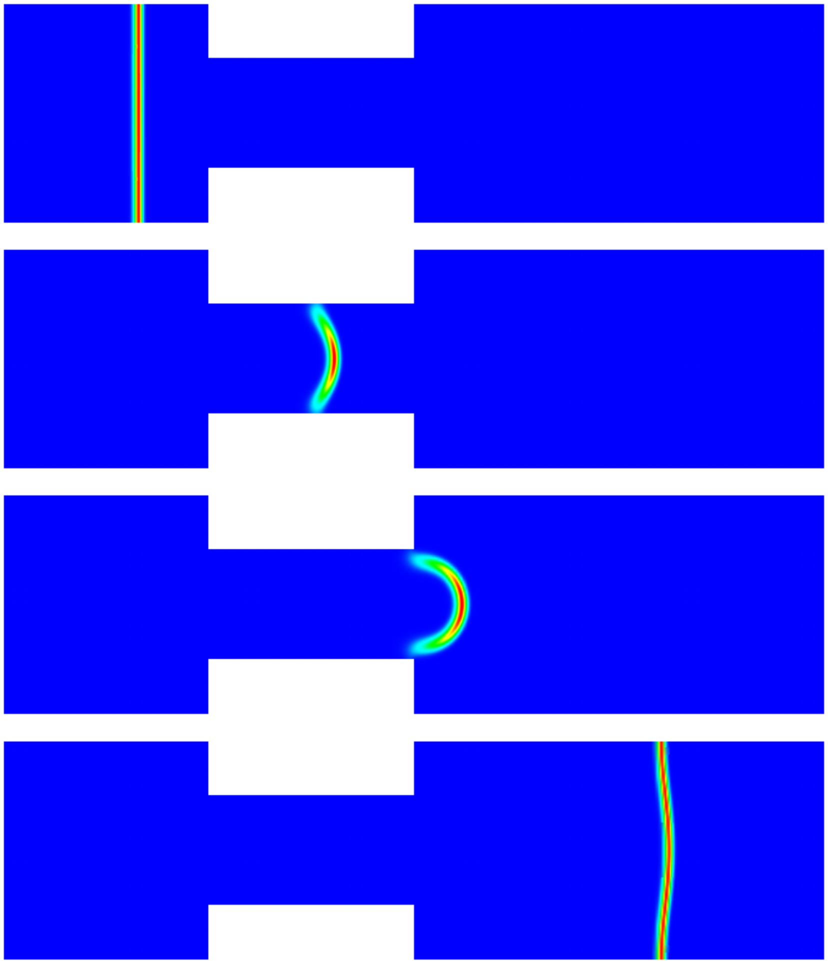


FIG. 5.4. *Example 1: Reaction rate ω at $t = 1.0$, 20.0, 40.0, 60.0.*

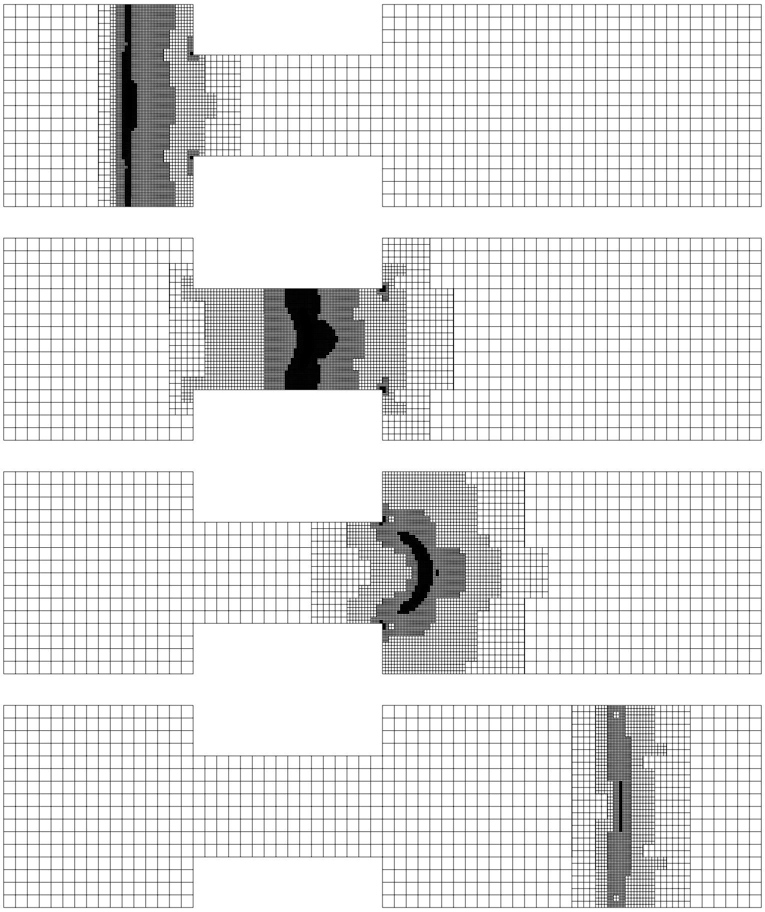


FIG. 5.5. *Example 1: Corresponding meshes at $t = 1.0$, 20.0, 40.0, 60.0.*

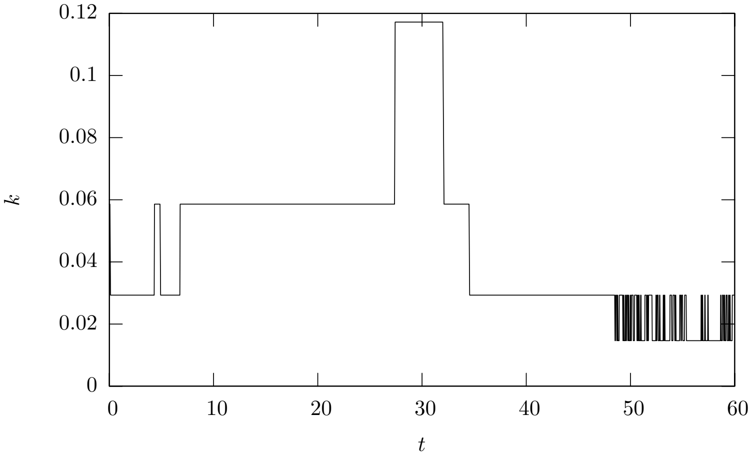


FIG. 5.6. *Example 1: Time steps k.*

TABLE 5.4
Example 2: Local refinement with equilibration for cG(1)dG(0) discretization.

M	$N_{\max}$	η_k	η_h	$\eta_k + \eta_h$	$J(u) - J(u_{kh})$	I_{eff}
256	985	$1.4{\cdot}10^{-02}$	$-2.3{\cdot}10^{-04}$	$1.416 \cdot 10^{-02}$	$2.681 \cdot 10^{-02}$	1.8925
342	985	$6.2{\cdot}10^{-03}$	$-1.1{\cdot}10^{-03}$	$5.149 \cdot 10^{-03}$	$2.038 \cdot 10^{-02}$	3.9574
466	985	$3.3{\cdot}10^{-03}$	$-2.1{\cdot}10^{-03}$	$1.294 \cdot 10^{-03}$	$1.681 \cdot 10^{-02}$	12.9945
748	1447	$2.2{\cdot}10^{-03}$	$4.6{\cdot}10^{-03}$	$6.776 \cdot 10^{-03}$	$1.078 \cdot 10^{-02}$	1.5912
1098	2357	$1.3{\cdot}10^{-03}$	$2.0{\cdot}10^{-03}$	$3.305 \cdot 10^{-03}$	$4.049 \cdot 10^{-03}$	1.2249
1446	4151	$9.4{\cdot}10^{-04}$	$6.4{\cdot}10^{-04}$	$1.577 \cdot 10^{-03}$	$1.715 \cdot 10^{-03}$	1.0876
1594	8655	$8.5{\cdot}10^{-04}$	$2.2{\cdot}10^{-04}$	$1.067 \cdot 10^{-03}$	$1.097 \cdot 10^{-03}$	1.0286

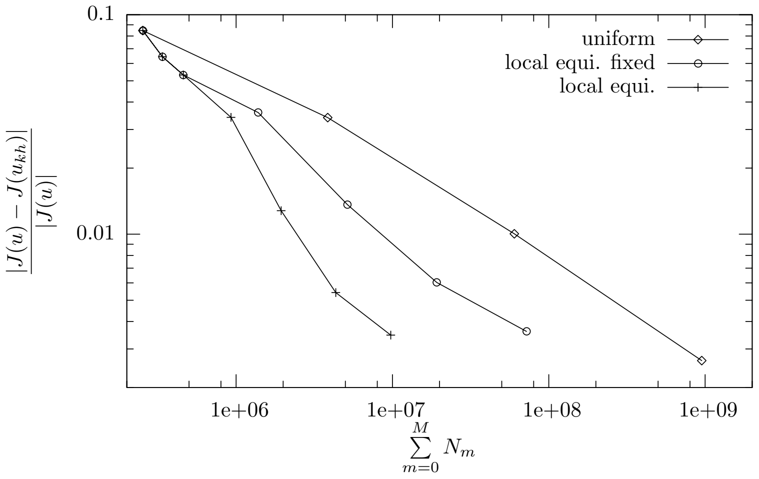


FIG. 5.7. *Example 2: Comparison of different refinement strategies for cG(1)dG(0) discretization.*

5.2. Example 2. For the second example, we consider the same problem as above but aim on reducing the error in a more localized functional:

$$J(u) := \frac{1}{60|\Gamma_{\mathrm{R}}|} \int_0^{60} \int_{\Gamma_{\mathrm{R}}} Y \, ds \, dt$$

describing the mean concentration of the species on the cooled rods Γ_{R}. The discretization is here done via the cG(1)dG(0) method. Table 5.4 again shows the development of the discretization error and the a posteriori error estimators during an adaptive run with local refinement in both types of discretizations. The notation is the same as in Tables 5.1 and 5.2. Further, we present in Figure 5.7 a comparison of the relative error $\frac{|J(u)-J(u_{kh})|}{|J(u)|}$ for the three refinement strategies.

In Figures 5.8 and 5.9, the spatial meshes and the size of the time steps are shown. Due to the more local structure of the functional J in this example, we observe a stronger benefit of our error estimator driven by the quantity of interest J than in Example 1. The spatial and temporal meshes are refined only if large residuals of the equation also have influence on the error measured in the functional J. This can be seen since the time steps at the end are significantly larger than at the beginning and the spatial meshes at the end are not refined at all.

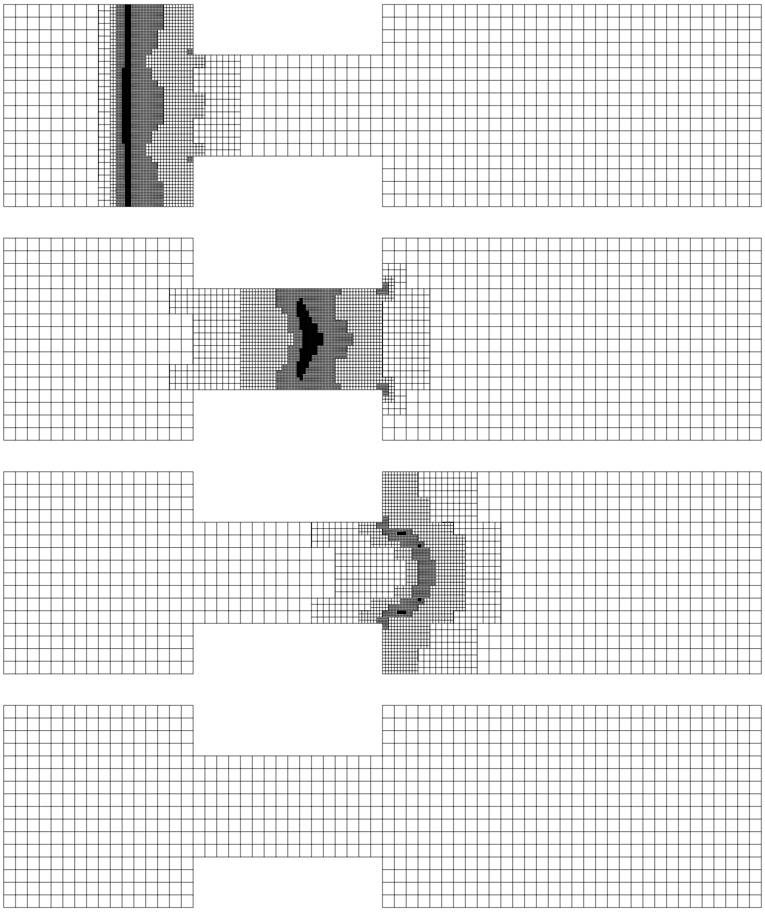


FIG. 5.8. *Example 2: Spatial meshes at $t = 1.0$, 20.0, 40.0, 60.0.*

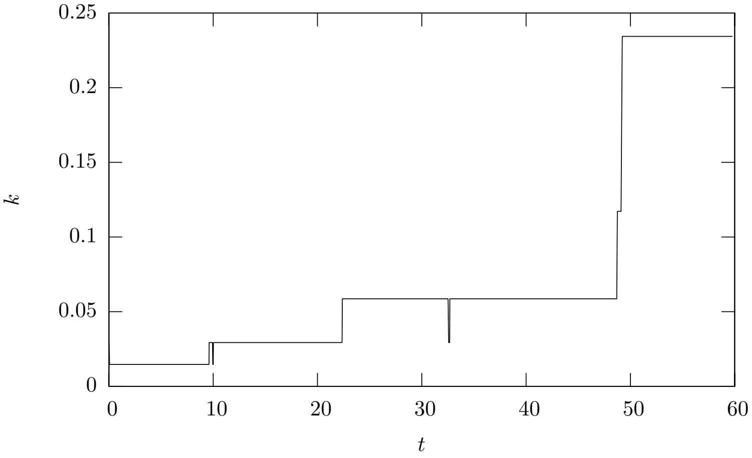


FIG. 5.9. *Example 2: Time steps k.*

6. Conclusion. In this paper we proposed a methodology for efficient adaptive choice of spatial and temporal discretizations for nonlinear parabolic problems. We derived an error estimator which assesses the discretization error with respect to a given quantity of interest describing the goal of the computation. This is an extension of the approach proposed in [4] to parabolic problems. The presented error estimator separates the discretization errors from the temporal and the spatial discretizations. This allows for equilibrating of these errors within an adaptive algorithm. Moreover, the spatial meshes corresponding to different time steps are refined separately, which is necessary for construction of efficient discretizations for problems with complex dynamic behavior. The presented algorithm is numerically tested and validated for a combustion problem under the low Mach number hypothesis.

REFERENCES

[1] G. Akrivis, C. Makridakis, and R. Nochetto, *A posteriori error estimates for the Crank-Nicolson method for parabolic equations*, Math. Comp., 75 (2006), pp. 511–531.

[2] R. Becker, D. Meidner, and B. Vexler, *Efficient numerical solution of parabolic optimization problems by finite element methods*, Optim. Methods Softw., 22 (2007), pp. 813–833.

[3] R. Becker and R. Rannacher, *A feed-back approach to error control in finite element methods: Basic analysis and examples*, East-West J. Numer. Math., 4 (1996), pp. 237–264.

[4] R. Becker and R. Rannacher, *An optimal control approach to a-posteriori error estimation*, in Acta Numerica 2001, Arieh Iserles, ed., Cambridge University Press, London, 2001, pp. 1–102.

[5] R. Becker, *Adaptive Finite Elements for Optimal Control Problems*, Habilitationsschrift, Institut für Angewandte Mathematik, Universität Heidelberg, 2003.

[6] Z. Chen and J. Feng, *An adaptive finite element algorithm with reliable and efficient error control for linear parabolic problems*, Math. Comp., 73 (2004), pp. 1167–1193.

[7] Ph. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, Classics Appl. Math. 40, SIAM, Philadelphia, 2002.

[8] R. Dautray and J.-L. Lions, *Mathematical Analysis and Numerical Methods for Science and Technology: Evolution Problems* I, Vol. 5, Springer-Verlag, Berlin, 1992.

[9] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to adaptive methods for differential equations*, in Acta Numerica 1995, Arieh Iserles, ed., Cambridge University Press, London, 1995, pp. 105–158.

[10] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Computational Differential Equations*, Cambridge University Press, Cambridge, 1996.

[11] K. Eriksson, C. Johnson, and A. Logg, *Adaptive computational methods for parabolic problems*, in Encyclopedia of Computational Mechanics, E. Stein, R. de Borst, and T. J. R. Hughes, eds., Wiley, New York, 2004, Vol. 1, Chap. 24.

[12] K. Eriksson, C. Johnson, and V. Thomée, *Time discretization of parabolic problems by the discontinuous Galerkin method*, RAIRO Modelisation Math. Anal. Numer., 19 (1985), pp. 611–643.

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

[14] 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., 32 (1995), pp. 706–740.

[15] J. Lang, *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems: Theory, Algorithm and Applications*, Lect. Notes Comput. Sci. Eng. 16, Springer-Verlag, Berlin, 2001.

[16] D. Meidner and B. Vexler, *Adaptive space-time finite element methods for parabolic optimization problems*, SIAM J. Control Optim., 46 (2007), pp. 116–142.

[17] M. Picasso, *Adaptive finite elements for a linear parabolic problem*, Comput. Methods Appl. Mech. Engrg., 167 (1998), pp. 223–237.

[18] 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., 38 (2000), pp. 837–875.

[19] R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Wiley/Teubner, New York/Stuttgart, 1996.