Available online at www.sciencedirect.com
SCIENCE DIRECT®
Applied Numerical Mathematics 54 (2005) 519–536
APPLIED NUMERICAL MATHEMATICS
www.elsevier.com/locate/apnum

Parameter identification for chemical models in combustion problems ☆

R. Becker [a], M. Braack [b,*], B. Vexler [b]

[a] *Laboratoire de Mathematiques Appliquees, Université de Pau et des Pays de l'Adour, BP 1155, 64013 Pau cedex, France*
[b] *Institute of Applied Mathematics, University of Heidelberg, INF 294, 69120 Heidelberg, Germany*

Abstract

We present an algorithm for parameter identification in combustion problems modeled by partial differential equations. The method includes local mesh refinement controlled by a posteriori error estimation with respect to the error in the parameters. The algorithm is applied to two types of combustion problems. The first one deals with the identification of Arrhenius parameters, while in the second one diffusion coefficients for an ozone flame are calibrated.

1. Introduction

Estimation of the unknown parameters in chemical models is indispensable for successful simulation and optimization of combustion problems. In this paper we present an algorithm for parameter identification in the context of multidimensional reactive flows. Typical problems are, for instance, the estimation of reaction rates or Arrhenius parameters and the estimation of diffusion coefficients. Since the arising system of partial differential equations is usually very complex, the solution of parameter identification

☆ This work has been supported by the German Research Foundation (DFG), through the *Sonderforschungsbereich 359* 'Reactive Flows, Diffusion and Transport', and the *Graduiertenkolleg* 'Complex Processes: Modeling, Simulation and Optimization' at the Interdisciplinary Center of Scientific Computing (IWR), University of Heidelberg.
* Corresponding author.
E-mail addresses: roland.becker@univ-pau.fr (R. Becker), malte.braack@iwr.uni-heidelberg.de (M. Braack), boris.vexler@iwr.uni-heidelberg.de (B. Vexler).

0168-9274/$30.00 © 2004 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.apnum.2004.09.017

problems requires development of special discretization and optimization techniques. In the presented algorithm we use local mesh refinement for finding efficient discretizations for parameter identification. The approach is based on a posteriori error estimation for the error in parameters and allows for reducing the dimension of the discretized problem to a minimum for achieving a prescribed accuracy. The optimization loop for determining the unknown parameters is intrinsically coupled with the mesh refinement algorithm: at the beginning, the optimization algorithm acts on coarse meshes. After achieving a sufficient reduction of the cost functional the mesh is refined and the optimization continues. These steps are iterated until the estimated error in parameters is below a user-specified tolerance. This allows to replace optimization iterations on fine meshes by iterations on coarse meshes. In addition, our algorithm includes the use of stabilized finite element discretizations on hierarchies of locally refined meshes, a multigrid procedure for solving the linear sub-problems, and a special optimization loop.

For discussing the subject, we consider the following simple model problem of a scalar stationary convection–diffusion–reaction equation (*cdr-equation*) for the variable u in a domain $\Omega \subset \mathbb{R}^2$ with a divergence-free vector field β and a diffusion coefficient D:

$$\beta \cdot \nabla u - \operatorname{div}(D\nabla u) + s(u, q) = f, \tag{1}$$

provided with Dirichlet boundary conditions $u = \hat{u}$ at the inflow boundary $\Gamma_{\text{in}} \subset \partial\Omega$ and Neumann conditions $\partial_n u = 0$ on $\partial\Omega \setminus \Gamma_{\text{in}}$. As usual in combustion problems, the reaction term is of Arrhenius type

$$s(u, q) := A \exp\bigl\{-E/(d-u)\bigr\} u(c-u). \tag{2}$$

While d, c are fixed parameters, the parameters A, E are considered as unknown and form the vector-valued parameter $q = (A, E) \in \mathbb{R}^2$. Since they are not directly measurable, we assume to have certain measurements $\overline{C} \in \mathbb{R}^{n_m}$, which should match with computed quantities $C(u) \in \mathbb{R}^{n_m}$. Here we may think, e.g., of laser measurements of mean concentrations along fixed lines, see Section 5. Calibration of Arrhenius parameter has be done by many scientists, for instance, by Lohmann [18] for coal pyrolysis which is frequently used in chemical engineering.

The aim of this work is the presentation of the numerical background of the proposed method and its validation by model problems. To this end numerical results for two test problems are presented. The first one deals with the estimation of Arrhenius parameters for one single reaction, as mentioned above (2). In the second example we analyze diffusion parameters in a combustion problem. For an overview of parameter estimation problems in chemistry, we refer to the book of Englezos and Kalogerakis [14]. Therein, many applications of parameter identification in the framework of ordinary differential equations are given. Parameter estimation problems for reactive flows in one space dimensions are treated, for instance, by Bock et al. [20].

This paper is organized as follows. In Section 2 we formulate the parameter identification problem as an optimization problem and describe the optimization algorithm for it on the continuous level. Thereafter, in Section 3 the optimization loop is applied to the stabilized finite element discretization of the problem. Section 4 is devoted to the adaptive mesh refinement algorithm and the a posteriori error estimation. In Section 5, the described algorithms are applied for estimating the Arrhenius coefficients of the scalar cdr-equation (1). A more complex ozone flame is analyzed in Section 6. It includes the equations for compressible flows, a system of cdr-equations for three chemical species and 6 elementary (bi-directional) reactions. The considered parameters calibrate a simple diffusion model in order to match with given observations. In both examples, the measurements $\overline{C}$ are produced numerically. In future work, the whole approach will be applied to examples with measured data coming from experiments.

2. The optimization algorithm for parameter identification problems

The aim of this section is the description of the optimization algorithms for the solution of the parameter identification problems. In Section 2.1, we start with the formulation of the parameter identification problem in variational form and describe the typical optimization algorithms for it on the continuous level including trust-region techniques.

2.1. The optimization algorithm for the continuous problem

We consider the parameter identification problem in the following variational form: the *state variable* u is supposed to be a sum of the function $\hat{u}$ describing the Dirichlet conditions and a function of a Hilbert space V, i.e., $u \in \widehat{V} := \hat{u} + V$. The unknown parameter q is assumed to be in the space $Q := \mathbb{R}^{n_p}$. The system of equations for the state variable u reads

$$a(u,q)(\phi) = (f,\phi) \quad \forall \phi \in V, \tag{3}$$

where $a(u,q)(\phi)$ is a form acting on the function space $\widehat{V} \times Q \times V$. This form is linear in ϕ and may be nonlinear in u and q. The right side f is assumed to be in the dual space V'. The form $a(u,q)(\phi)$ is assumed to be differentiable with respect to u and q with Gateaux derivatives a'_u and a'_q, respectively.

For the cdr-equation (1) the form $a(u,q)(\phi)$ is obtained by multiplication of Eq. (1) with test functions $\phi \in V = \{v \in H^1(\Omega) | v_{\Gamma_{\text{in}}} = 0\}$, integration over the computational domain Ω and integration by parts of the diffusive term. The resulting discrete Galerkin form is

$$a(u,q)(\phi) := \int_\Omega \big(\beta \cdot \nabla u + s(u,q)\big)\phi \,\mathrm{d}x + \int_\Omega D \nabla u \nabla \phi \,\mathrm{d}x. \tag{4}$$

Further, the measurable quantities are represented by a linear observation operator $C : \widehat{V} \to Z$, which maps the state variable u into the space of measurements $Z := \mathbb{R}^{n_m}$ with $n_m \geqslant n_p$. We denote by $\langle \cdot, \cdot \rangle_Z$ the Euclidean scalar product of Z and by $\| \cdot \|_Z$ the corresponding norm. Similar notations are used for the scalar product and norm in the space Q.

The values of the parameters are estimated from a given set of measurements $\overline{C} \in Z$ using a least-squares approach, in such a way that we obtain the constrained optimization problem with the cost functional $J : \widehat{V} \to \mathbb{R}$:

$$\text{Minimize } J(u) := \frac{1}{2}\big\|C(u) - \overline{C}\big\|_Z^2, \text{ under the constraint (3).} \tag{5}$$

Under a regularity assumption for a'_u, the implicit function theorem in Banach spaces, see Dieudonné [13], implies the existence of an open set $Q_0 \subset Q$, containing the optimal parameter q, and a continuously differentiable solution operator $S : Q_0 \to V$, $q \to S(q)$, so that (3) is fulfilled for $u = S(q)$. Using this solution operator S, we define the reduced observation operator $c : Q_0 \to Z$ by $c(q) := C(S(q))$, in order to reformulate the problem under consideration as an unconstrained optimization problem with the reduced cost functional $j : Q_0 \to \mathbb{R}$:

$$\text{Minimize } j(q) := \frac{1}{2}\big\|c(q) - \overline{C}\big\|_Z^2, \quad q \in Q_0. \tag{6}$$

Denoting by $G = c'(q)$ the Jacobian matrix of the reduced observation operator c, the first-order necessary condition $j'(q) = 0$ for (6) reads

$$G^*\big(c(q) - \overline{C}\,\big) = 0, \tag{7}$$

where G^* denotes the transpose of G. The unconstrained optimization problem (6) is solved iteratively. Starting with an initial guess q^0, the next parameter is obtained by $q^{k+1} = q^k + \delta q$, where the update δq is the solution of the problem

$$H_k \delta q = G_k^* r_k, \quad \text{where } r_k := \overline{C} - c\big(q^k\big), \quad G_k := c'\big(q^k\big), \tag{8}$$

and H_k is an approximation of the Hessian $\nabla^2 j(q^k)$ of the reduced cost functional j. Although δq depends on the iterate k, we suppress the index in order to facilitate the readability. The choice of the matrix $H_k \in \mathbb{R}^{n_p \times n_p}$ leads to different variants of the optimization algorithm. We consider the following typical possibilities:

2.1.1. Gauß–Newton algorithm

The choice $H_k := G_k^* G_k$ corresponds to the Gauß–Newton algorithm, which can be interpreted as the solution to the linearized minimization problem

$$\text{Minimize } \frac{1}{2}\big\|c\big(q^k\big) + G_k \delta q - \overline{C}\,\big\|^2. \tag{9}$$

The components G_{ij} of the Jacobian G_k can be computed as follows:

$$G_{ij} := \frac{\partial c_i}{\partial q_j}\big(q^k\big) = C_i(w_j), \quad i = 1, \ldots, n_m,\ j = 1, \ldots, n_p,$$

where C_i and c_i denote the components of the observation and the reduced observation operators, respectively. The tanget solution $w_j \in V$ is determined by

$$a'_u\big(u^k, q^k\big)(w_j, \phi) = -a'_{q_j}\big(u^k, q^k\big)(\phi) \quad \forall \phi \in V,\ j = 1, \ldots, n_p, \tag{10}$$

where $u^k = S(q^k)$. For one Gauß–Newton step, the state equation (3) for $u^k = S(q^k)$ and n_p tangent problems (10) have to be solved which originate from the same linear operator but with different right sides. The solution of (8) is uncritical due to the small dimension of H_k. Note that we suppress the index k for the matrix entries G_{ij} and the vectors w_j.

2.1.2. Full Newton algorithm

Another possibility is to set $H_k := \nabla^2 j(q^k)$, which leads to the full Newton algorithm. The required Hessian $\nabla^2 j(q^k)$ is given by

$$\nabla^2 j\big(q^k\big) = G_k^* G_k + M_k. \tag{11}$$

As before, the computation of the Jacobian G_k is required. The entries of the matrix $M_k \in \mathbb{R}^{n_p \times n_p}$ can be computed by a subtle evaluation of several second derivatives of the form $a(u, q)(\phi)$ in the directions w_j (the solutions of the tangent problems (10)) and $z \in V$, the solution of the adjoint equation

$$a'_u\big(u^k, q^k\big)(\phi, z) = -(r_k, C\phi) \quad \forall \phi \in V. \tag{12}$$

Since we do not use this method for the problems under consideration, we refer to Becker and Vexler [4] for details. For convergence theory of Gauß–Newton and Newton methods see, e.g., Dennis and Schnabel [11] or Nocedal and Wright [19].

For large least-squares residuals $\|C(u) - \overline{C}\|_Z$, the Gauß–Newton algorithm often shows slow convergence. The full Newton algorithm has better (local) convergence properties, because it leads to quadratic convergence. However, for reactive flow problems, the evaluation of the second derivatives of $a(u, q)(\phi)$ is usually very expensive. Therefore, the use of the full Newton algorithms is often unattractive, or even impossible. We discuss shortly an alternative algorithm which combines the comparative "low" cost of the Gauß–Newton method and the better convergence properties of the full Newton.

2.1.3. Gauss–Newton-update method

Based on the ideas of Dennis et al. [12], we replace the expensive matrix M_k in (11) by an approximation obtained by an update formula. It produces a sequence of computable matrices $\widehat{M}_k$. Starting with $\widehat{M}_0 = 0$:

$$\widehat{M}_{k+1} = \widehat{M}_k + \frac{1}{y^*\delta q}\left(xy^* + yx^*\right) - \frac{x^*\delta q}{(y^*\delta q)^2}yy^*,$$

where $y = G^*_{k+1}r_{k+1} - G^*_k r_k$, $x = (G^*_{k+1} - G^*_k)\,r_{k+1} - \widehat{M}_k\delta q$. Then, for the matrix H_k we use the following Hessian approximation: $H_k := G^*_k G_k + \widehat{M}_k$. Note, that no further equations have to be solved for the determination of $\widehat{M}_k$.

The matrices $\widehat{M}_k$ are chosen in such a way that H_k is a secant approximation of the (exact) Hessian. For derivation and analysis of this update formula, see also [11]. In Section 6 we compare this algorithm with the Gauß–Newton method and observe a substantial difference in the required number of iterations.

2.2. Trust-region method

It is well known, that the convergence of the algorithms described so far is ensured, only if the initial guess q^0 is in a sufficiently small neighborhood of the optimal parameter q. We use trust-region techniques in order to improve the global convergence, see, e.g., [11] or [19]. In the following, we shortly describe this algorithm. If the matrix H_k is positive definite, the computation of $\delta q \in Q$ in (8) can be interpreted as the solution of a minimization problem (cf. (9)):

$$\text{Minimize } m_k(\delta q) := j\left(q^k\right) - r_k^* G_k\delta q + \frac{1}{2}\delta q^* H_k\delta q, \quad \delta q \in Q. \tag{13}$$

The cost functional m_k of (13) is the so-called *local model function*, which behavior near the current point q^k is similar to that of the actual cost functional j defined in (6). However, the local model function m_k may not be a good approximation of j for large δq. Therefore, we restrict the search for a minimizer of m_k to a ball (*trust region*) around q^k. In other words, we replace the problem (13) by the following constrained optimization problem:

$$\text{Minimize } m_k(\delta q), \quad \text{subject to } \|\delta q\|_Q \leqslant \Delta_k, \tag{14}$$

with a trust-region radius Δ_k to be determined iteratively.

For the convergence properties of the trust-region method, the strategy for choosing the trust-region radius Δ_k is crucial. Following the standard approach, see, e.g., Conn et al. [10], we base this choice on the agreement between the model function m_k and the cost functional j at the previous iteration. For the increment δq, we define the ratio

$$\rho_k = \frac{j(q^k) - j(q^k + \delta q)}{m_k(0) - m_k(\delta q)},$$

and use it as an indicator of the quality of the local model m_k. If this ratio is close to 1, there is a good agreement between the model m_k and the cost functional j for the current step. As a consequence, the trust region is expanded for the next iteration. Otherwise, we do not alter the trust region or shrink it, depending on the distance $|\rho_k - 1|$, see [19] for a precise definition.

The solution of the quadratic minimization problem (14) requires an additional remark. Due to the compactness of the feasible set described by the condition $\|\delta q\|_Q \leqslant \Delta_k$, the problem (14) posses always a solution independently of the definiteness of the matrix H_k. If the matrix H_k is positive definite and it holds

$$\left\| H_k^{-1} G_k^* r_k \right\|_Q \leqslant \Delta_k,$$

we set $\delta q = H_k^{-1} G_k^* r_k$. Otherwise, the solution δq is searched on the boundary of the feasible set $\{\delta q | \|\delta q\|_Q \leqslant \Delta_k\}$, and is determined by

$$\delta q = (H_k + \lambda I)^{-1} G_k^* r_k,$$

where I is the identity matrix and $\lambda > 0$ is chosen, such that $\|\delta q\|_Q = \Delta_k$. For computation of λ, the singular value decomposition of H_k is computed and λ is determined by the scalar equation, which is solved by one-dimensional Newton method, see, e.g., [19] for details.

For the numerical examples in Sections 5 and 6, the optimization algorithm does not converge without using such globalization techniques.

3. The discretization by finite elements

The continuous state equation (3) and several tangent problems (10) have to be discretized. The easiest possibility is to replace these equations by some numerical discrete approximations on a "sufficient fine" mesh resulting from the uniform refinement of the starting mesh. However, the naturally arising questions here are: first, how to decide if a mesh is sufficient fine? Second, are the meshes produced by the uniform refinement economical for the computation of parameters? And third, how to design another mesh refinement procedure in order to obtain more efficient meshes? These questions are extremely important in combustion problems, because of arising thin flame fronts. Furthermore, in parameter estimation problems the measurements are usually local quantities which gives the need of local mesh refinement. The required procedure is described in Section 4.

3.1. Meshes and finite element spaces

For the discretization we use a conforming equal-order Galerkin finite element method defined on quadrilateral meshes $\mathcal{T}_h = \{K\}$ over the computational domain $\Omega \subset \mathbb{R}^2$, with cells denoted by K. The mesh parameter h is defined as a cell-wise constant function by setting $h|_K = h_K$ and h_K is the diameter of K. The straight parts which make up the boundary ∂K of a cell K are called *faces*.

A mesh $\mathcal{T}_h$ is called regular, if it fulfills the standard conditions for shape-regular finite element mesh, see, e.g., Ciarlet [8]. However, in order to easy the mesh refinement we allow the cell to have nodes, which lie on midpoints of faces of neighboring cells. But at most one such *hanging node* is permitted for each face.

The discrete function space $V_h \subset V$ consist of continuous, piecewise polynomial functions (so-called Q_1-elements) for all unknowns,

$$V_h = \{\varphi_h \in C(\overline{\Omega});\ \varphi_h|_K \in Q_1\},$$

where Q_1 is the space of functions obtained by transformations of (isoparametric) bilinear polynomials on a fixed reference unit cell $\widehat{K}$. For a detailed description of this standard construction, see [8] or Johnson [17].

The case of hanging nodes requires some additional remarks. There are no degrees of freedom corresponding to these irregular nodes and the value of the finite element function is determined by pointwise interpolation. This implies continuity and therefore global conformity, i.e., $V_h \subset V$. For implementation details, see, e.g., Carey and Oden [7].

For several applications, the Galerkin formulation is not stable. For instance, at higher Reynolds numbers, advective terms become unstable. In order to overcome this limitation, stabilization techniques can be used. For this, the triangulation $\mathcal{T}_h$ is supposed to be constructed in such a way that it results from a coarser quasi-regular mesh $\mathcal{T}_{2h}$ by one global refinement. By a "patch" of elements we denote a group of four cells in $\mathcal{T}_h$ which results from a common coarser cell in $\mathcal{T}_{2h}$. The corresponding discrete finite element spaces V_{2h} and V_h are nested: $V_{2h} \subset V_h$. By I_{2h}^h we denote the nodal interpolation operator $I_{2h}^h : V_h \to V_{2h}$. By

$$\pi_h : V_h \to V_h, \qquad \pi_h \xi = \xi - I_{2h}^h \xi$$

we denote the difference between the identity and this interpolation.

3.2. The stabilized nonlinear form

Let the discrete function spaces be given by $V_h \subset V$, $\widehat{V}_h := \hat{u}_h + V_h$, with an approximation $\hat{u}_h$ of the boundary data $\hat{u}$. For fixed parameter $q_h \in Q$, the discrete solution $u_h \in \widehat{V}_h$ is determined by the discretized state equation

$$a_h(u_h, q_h)(\phi) = (f, \phi) \quad \forall \phi \in V_h. \tag{15}$$

The nonlinear form $a_h(u_h, q_h)(\phi)$ results from a stabilized finite element discretization on the mesh $\mathcal{T}_h$, given by a sum of the Galerkin part and stabilization terms:

$$a_h(u_h, q_h)(\phi) = a(u_h, q_h)(\phi) + b_h(u_h, q_h)(\phi).$$

We note that the kind of the discretization is not essential for the discrete optimization loop described below. However, the use of finite elements is crucial for the derivation of the a posteriori error estimation in Section 4.

3.2.1. Convection stabilization

For a cdr-equation (1), the stabilization term added to the Galerkin formulation reads

$$b_h(u_h, q_h)(\phi) := \sum_{K \in \mathcal{T}_h} \delta_K \int_K (\beta \cdot \nabla \pi_h u_h)(\beta \cdot \nabla \pi_h \phi)\, \mathrm{d}x, \tag{16}$$

where the cell-wise coefficients δ_K depend on the local balance of convection and diffusion:

$$\delta_K := \frac{\delta_0 h_K^2}{6D + h_K \|\beta\|_K}.$$

Here, the quantities h_K and $\|\beta\|_K$ are cell-wise values for the cell-size and the convection β. The parameter δ_0 is a fixed constant, usually chosen as $\delta_0 = 0.5$. Note, that π_h vanishes on V_{2h}, and therefore, the stabilization vanishes for test functions of the coarse grid $\xi \in V_{2h}$. This type of stabilization is analyzed by Guermond [15].

For systems of cdr-equations, for each convective term, one stabilization term of type (16) is added. The convection β and the particular diffusion coefficient D may depend itself from u and may be different for each sub-equation.

3.2.2. Pressure-velocity stabilization

For equal-order finite elements, the Galerkin formulation of the Stokes system for the pressure p and velocity v,

$$\operatorname{div} v = 0, \qquad -\mu \Delta v + \nabla p = f$$

is known to be unstable, since the stiff pressure-velocity coupling for (nearly) incompressible flows enforces spurious pressure modes. The same occurs for hydrodynamic incompressible flows since they involve also the saddle-point structure of the Stokes system. Let p_h denote the discrete pressure, v_h the discrete velocity, $u_h = (p_h, v_h)$, and ξ the test function for the divergence equation. The added stabilization term which damps acoustic pressure modes is of the form

$$b_h(u_h, q_h)(\phi) = \sum_{K \in \mathcal{T}_h} \alpha_K \int_K (\nabla \pi_h p_h)(\nabla \pi_h \xi)\, dx, \tag{17}$$

with weights $\alpha_K = \alpha_0 h_K^2/\mu$ depending on the mesh size h_K of cell K and the viscosity μ. The parameter α_0 is usually chosen between 0.2 and 1. The stabilization term (17) acts as a diffusion term on the fine-grid scales of the pressure. The scaling proportional to h_K^2 give stability of the discrete equations and maintain accuracy. This type of stabilization is introduced in Becker and Braack [1] for the Stokes equation. Therein, a stability proof and an error analysis is given. The same stabilization is applied to the (compressible) Navier–Stokes equations.

The proposed stabilization is consistent in the sense that the introduced terms vanish for $h \to 0$. For smooth solutions, the introduced perturbation is even of higher-order than the discretization error.

3.3. The finite-dimensional optimization algorithm

Analog to Section 2.1, we assume the regularity of the derivative $(a_h)'_u$, which implies the existence of a discrete solution operator S_h, such that $u_h = S_h(q_h)$ fulfills the discrete state equation (15). Moreover, we introduce the discrete reduced observation operator c_h by setting $c_h(q_h) = C(S_h(q_h))$. The optimization loop on a given mesh $\mathcal{T}_h$ for the problem

$$\text{Minimize } j_h(q_h) := \frac{1}{2}\big\|c_h(q_h) - \overline{C}\big\|_Z^2, \quad q_h \in Q,$$

starts with an initial guess for the parameters $q_h^0 \in Q$. Thereafter, the corresponding discrete state u_h^k and the next parameter q_h^{k+1} are obtained by the discrete equations

$$u_h^k \in \widehat{V}_h\colon a_h\big(u_h^k, q_h^k\big)(\phi) = (f, \phi) \quad \forall \phi \in V_h,$$
$$\delta q_h \in Q\colon H_h \delta q_h = G_h^* r_h, \quad r_h := \overline{C} - c_h\big(q_h^k\big),$$

$$q_h^{k+1} \in Q\colon q_h^{k+1} = q_h^k + \delta q_h, \tag{18}$$

where $G_h := c_h'(q_h^k)$ and H_h the discrete approximation of H_k according the choice above. The globalization technique formulated for the continuous problems in Section 2.2 can be carried over to the discrete case analogously.

4. Adaptive mesh refinement via a posteriori error estimation

In this section, we describe the adaptive algorithm for mesh refinement and error control based on the a posteriori error estimation for parameter identification problems developed in [4]. In order to measure the error in the parameters, we introduce an error functional $E : Q \to \mathbb{R}$. The use of the error functional E allows to weight the relative importance of the different parameters. The following error representation holds:

$$E(q) - E(q_h) = \eta_h + P + R. \tag{19}$$

Here, η_h denotes the computable a posteriori error estimator given later. The remainder term R is due to linearization and is cubic in the error. The term P appears if an inexact Newton algorithm (e.g., Gauß–Newton) is applied. However, P vanishes if the optimal parameters perfectly match for (6), i.e., $j(q) = 0$. Since the parts P and R are omitted in the numerical examples in this work, we simply refer to [4] for details.

The error estimator is based on the optimal control approach to a posteriori error estimation developed in Becker and Rannacher [2,3]. However, a direct application of this approach leads to an estimator which controls the error in the cost functional (5). In general, such an estimator does not provide useful error bounds for the parameters, in contrast to the estimator (19) described in the following.

We sketch a generic adaptive mesh refinement algorithm. Such an algorithm generates a sequence of locally refined meshes and corresponding finite element spaces until the estimated error is below a given tolerance *TOL*. For the following iteration, we have a mesh refinement procedure that adaptively refines a given regular mesh to obtain a new regular mesh for the next iteration. The refinement procedure is guided by information based on the cell-wise contributions of the estimator η_h.

Adaptive mesh refinement algorithm

1. Choose an initial mesh $\mathcal{T}_{h_0}$ and set $l = 0$
2. Construct the finite element space V_{h_l}
3. Compute the discrete optimal $q_{h_l} \in Q$, i.e., iterate (18)
4. Evaluate the a posteriori error estimator η_{h_l}
5. If $\eta_{h_l} \leqslant TOL$ quit
6. Refine $\mathcal{T}_{h_l} \to \mathcal{T}_{h_{l+1}}$ using information from η_{h_l}
7. Increment l and go to 2

In step 3 the least-squares problem is solved on a fixed mesh. As initial data we use the values from the computation on the previous mesh. This allows us to avoid unnecessary iterations of the optimization loop on fine meshes.

For evaluation of our a posteriori error estimator η_h, we consider an additional adjoint equation for the adjoint variable $y \in V$:

$$a'_u(u,q)(\phi,y) = \left\langle G\left(G^*G\right)^{-1}\nabla E(q), C(\phi)\right\rangle_Z \quad \forall \phi \in V, \tag{20}$$

and solve the discrete version of it, i.e., $y_h \in V_h$:

$$(a_h)'_u(u_h,q_h)(\phi,y_h) = \left\langle G_h\left(G_h^*G_h\right)^{-1}\nabla E(q_h), C(\phi)\right\rangle_Z \quad \forall \phi \in V_h. \tag{21}$$

We denote by ρ and ρ^* the residuals of the state and the adjoint equations, respectively, i.e., we define for test functions $\phi \in V$:

$$\rho(u_h)(\phi) := (f,\phi) - a_h(u_h,q_h)(\phi),$$
$$\rho^*(u_h,y_h)(\phi) := \left\langle G_h\left(G_h^*G_h\right)^{-1}\nabla E(q_h), C(\phi)\right\rangle_Z - (a_h)'_u(u_h,q_h)(\phi,y_h).$$

Using this notation, the error estimator is given by

$$\eta_h = \frac{1}{2}\rho(u_h)(y - i_h y) + \frac{1}{2}\rho^*(u_h,y_h)(u - i_h u), \tag{22}$$

where $i_h : V \to V_h$ is an appropriate interpolation operator, see Clément [9]. For simplicity we assume that $\hat{u}_h = \hat{u}$, such that $u - i_h u \in V$. For a proof of (19) with the error estimator given by (22), see [4].

For evaluation of this error estimator, the local interpolation errors $y - i_h y$ and $u - i_h u$ have to be approximated. In our numerical examples, we use interpolation of the computed bilinear finite element solutions y_h and u_h on the space of biquadratic finite elements on patches of cells.

The main computational cost for the a posteriori error estimator described above is the solution of one auxiliary equation (21). This is cheap, even in comparison with only one Gauß–Newton step, which includes solution of the state (nonlinear) and of the several (linear) tangent equations.

These residual terms are still global quantities. In order to use it for local mesh adaptation, the estimator η_h has still to be localized to cell-wise or node-wise error indicators. For the numerical results in this work, we perform node-wise localization by summation over all nodes of the mesh. For a mesh $\mathcal{T}_h$ with N nodes, the estimator can be expressed by $\eta_h = \sum_{i=1}^{N} \tau_i$. Then, the mesh is locally refined with respect to the error indicators $\eta_i := |\tau_i|$. For more details on the localization procedure used, we refer to Braack and Ern [6]. However, there are also methods for localization to cell-wise quantities, see [3].

5. Identification of Arrhenius parameters

The first example we analyze with respect to the proposed optimization algorithm is the scalar cdr-equation (1) with $f \equiv 1$, $D = 10^{-6}$, and a chemical source term of Arrhenius type (2). The variable u stands for the mole fraction of a fuel, while the mole fraction of the oxidizer is $0.2 - u$. Since the Arrhenius law is a heuristic law and cannot be derived by physical laws, the involved parameters are a priori unknown and have to be calibrated. This parameter fitting is usually done by comparison of experimental data and simulation results. Therefore, this example is well suited for the proposed parameter identification algorithm.

Fuel (F) and oxidizer (Ox) are injected in different pipes and diffuse in a reaction chamber with overall length 35 mm and height 7 mm, see Fig. 1. At the center tube, the Dirichlet condition for the

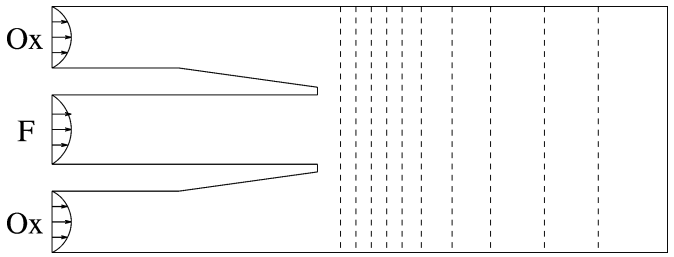

Fig. 1. Configuration of the reaction chamber for estimating Arrhenius coefficients. Dashed vertical lines indicate schematically the lines where the measurements are modeled.

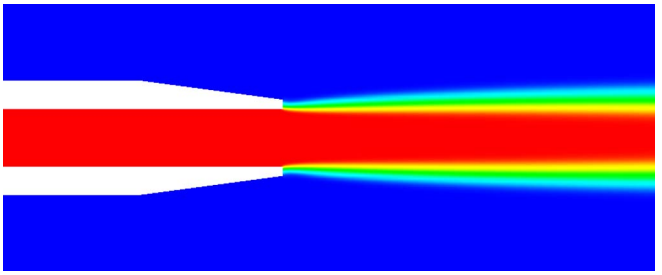

Fig. 2. Mole fraction of the fuel (u^0) for the initial parameters q^0.

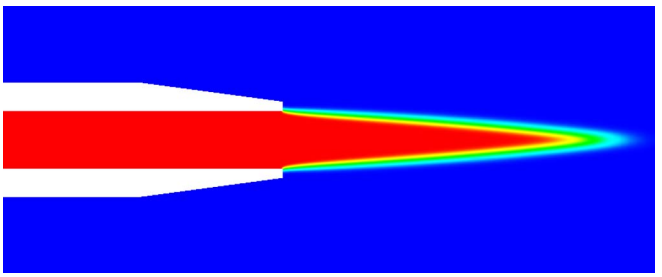

Fig. 3. Mole fraction of the fuel (u) for the optimal parameters q (right).

fuel is $u = u_{\text{in}} := 0.2$, and at the upper and lower tube, $u = 0$. On all other parts of the boundary, homogeneous Neumann conditions are opposed. The fix parameters in the Arrhenius law (2) are $c = u_{\text{in}}$ and $d = 0.24$. The convection direction $\beta(x, y)$ is a velocity field obtained by solving the incompressible Navier–Stokes equations with parabolic inflow profile at the tubes with peak flow $\beta_{\max} = 0.2$ m/s. The initial parameters are set to $q^0 = (\log(A^0), E^0) = (4, 0.15)$, leading to low reaction rates and a diffusion dominated solution. In Fig. 2 the corresponding state variable (fuel) u^0 is shown.

We simply choose the optimal parameters to $q = (6.9, 0.07)$ and replace the measurements by computations with these parameters: $\overline{C} := C(S(q))$. For this, we use a very fine locally refined mesh with more than 100 000 nodes. As a consequence, on this mesh the "measurements" perfectly match for the optimal parameters. This will not be the case in the second example. The state variable $u = S(q)$ is shown in Fig. 3. For the optimal q, in contrast to the initial guess q^0, a sharp reaction front occurs. Obviously, the difference in the parameters has a substantial impact to the state u. The measurements $C(u) \in \mathbb{R}^{n_m}$ are

Table 1
Numerical results for Arrhenius parameter identification

N	it	$\|C(u) - \overline{C}\|$	Res	q_1	q_2
1664	1	5.87e−2	2.54e−3	4.000	1.500e−1
	2	5.86e−2	2.56e−3	4.001	1.499e−1
	3	5.81e−2	2.83e−3	4.132	1.499e−1
	4	4.47e−2	6.42e−3	5.630	1.489e−1
	5	2.36e−2	5.58e−3	6.752	1.481e−1
	6	6.22e−3	1.90e−3	7.433	1.093e−1
	7	8.34e−4	2.55e−4	6.660	4.621e−2
	8	3.00e−4	1.00e−4	6.825	6.394e−2
2852	1	4.79e−4	1.44e−4	6.798	6.216e−2
	2	3.68e−5	1.04e−5	6.905	7.134e−2
	3	1.92e−5	5.51e−8	6.906	7.158e−2
6704	1	2.33e−4	7.72e−5	6.906	7.158e−2
	2	1.42e−5	1.91e−8	6.904	7.066e−2
13 676	1	6.91e−5	2.30e−5	6.904	7.063e−2
	2	3.53e−6	6.76e−9	6.905	7.052e−2
21 752	1	1.22e−5	3.24e−6	6.905	7.052e−2
	2	2.84e−6	8.89e−9	6.902	7.022e−2

modeled by mean values along $n_m = 10$ straight lines Γ_i at different positions in the reaction chamber, see dashed lines in Fig. 1, i.e.,

$$C_i(v) = \int_{\Gamma_i} v \,\mathrm{d}x, \quad i = 1, \ldots, n_m.$$

For the error functional, we choose the discretization error with respect to the second parameter $E(q) = q_2$. In the optimization loop, we use the Gauß–Newton algorithm with the trust-region strategy described before. In Table 1, the results obtained are listed. The third column displays the corresponding cost functional. On the first mesh with only $N = 1664$ nodes, 8 iterations (see second column) are done. On this mesh, the cost functional is reduced by more than two digits. In the fourth column, the remaining residual of the optimization condition (7) (in the discrete form) is listed:

$$\text{Res} := \left\| G_h^*\left(\overline{C} - c_h\left(q_h^k\right)\right)\right\|. \tag{23}$$

The last two columns show the corresponding obtained parameters. After a substantial reduction of Res, the mesh is adapted locally according the a posteriori error estimator η_h. The second mesh has 2852 nodes. Here, the optimization loop is repeated. However on the finer meshes, only a few ($\leqslant 3$) iterations are necessary. On the finest mesh, the error in the first parameter is about 0.03% and in the second parameter about 0.3%.

Comparing the error in the parameters with a more conventional strategy on globally refined meshes, our proposed algorithm is much more efficient. In Fig. 4, the absolute difference in the second parameter is plotted in dependence of the number of mesh points. The dashed line results from our method on locally refined meshes. The solid line stands for parameter estimation with the same optimiza-

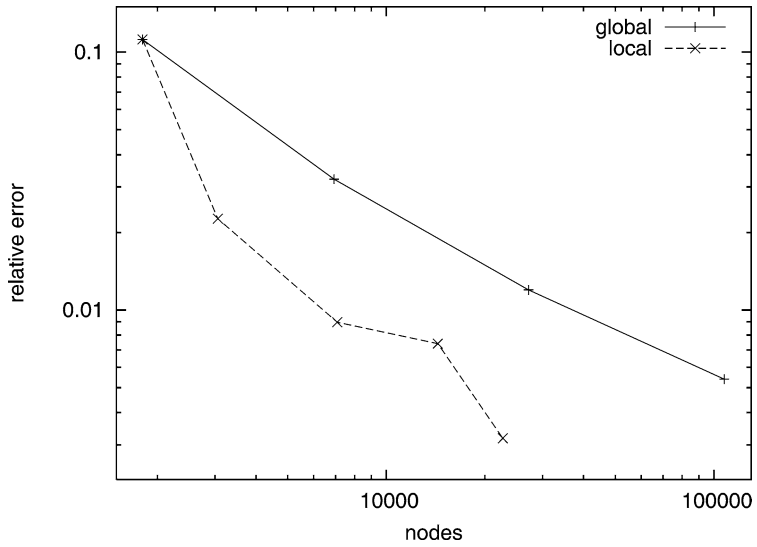

Fig. 4. Relative error in the second Arrhenius parameter in dependence of the number of mesh points. Solid line: globally refined meshes; dashed line: locally refined meshes on the basis of a posteriori error estimation.

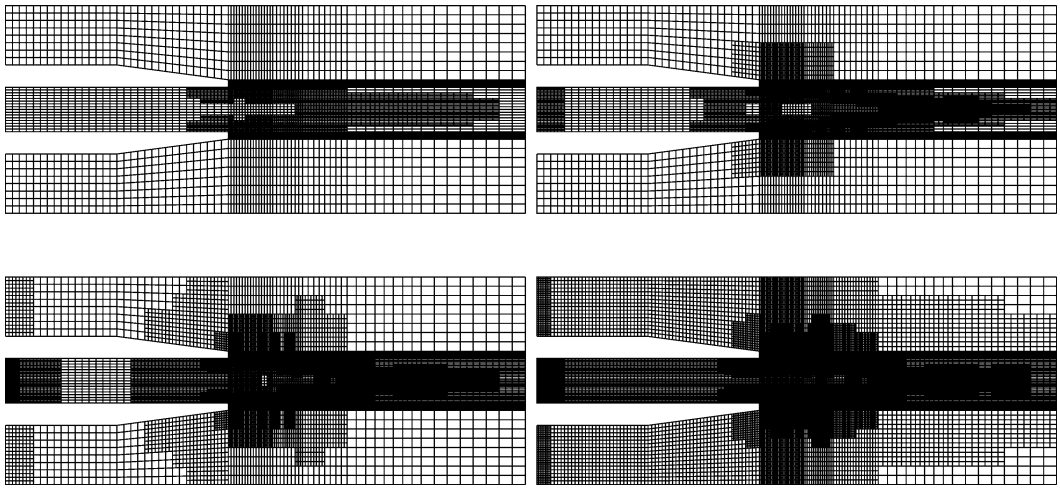

Fig. 5. Obtained meshes for estimating Arrhenius parameters with 2852, 6704, 13 676 and 21 752 nodes (from upper left to lower right).

tion loop but on tensor meshes. For a relative error of less than 1%, only 6704 nodes are necessary with a locally refined mesh, whereas more than 100 000 nodes are necessary on a uniformly refined mesh.

In Fig. 5, a sequence of locally refined meshes produced by the refinement algorithm is shown. The highest amount of mesh points is located near the flame front and close to the measurement lines.

6. Identification of diffusion parameters

In this example, we consider a stationary ozone flame modeled by the following system of equations for velocities v, pressure p, temperature T and mass fractions y_k:

$$\operatorname{div}(\rho v) = 0, \qquad (\rho v \cdot \nabla)v + \operatorname{div}\pi + \nabla p = 0,$$
$$\rho v \cdot \nabla T - \frac{1}{c_p}\operatorname{div}\mathcal{Q} = -\sum_{i\in\mathcal{S}} h_i f_i, \qquad \rho v \cdot \nabla y_k + \operatorname{div}\mathcal{F}_k = f_k, \quad k \in \mathcal{S}.$$

The specific enthalpies are denoted by h_i, the heat capacity at constant pressure is denoted by c_p. Both quantities are evaluated by the use of thermodynamic data bases. The set $\mathcal{S}$ denotes the set of chemical species. The density ρ is given by the perfect gas law in a mixture with partial molecular weights m_i and the uniform gas constant R:

$$\rho = \frac{p}{RT}\left(\sum_{i\in\mathcal{S}} \frac{y_i}{m_i}\right)^{-1}.$$

The stress tensor π is given as usual for compressible flows. The reaction terms f_i are modeled by Arrhenius laws for reactions with reaction rates k_r:

$$f_i = m_i \sum_{r\in\mathcal{R}} (\nu'_{ri} - \nu_{ri}) k_r \prod_{s\in\mathcal{S}} c_j^{\nu_{rj}}, \quad k_r = A_r T^{\beta_r} \exp\left\{-\frac{E_r}{RT}\right\}.$$

The set $\mathcal{R}$ includes all reactions considered. The stoichiometric coefficients of the products and educts for reaction r are denoted by ν'_{ri} and ν_{ri}, respectively. The concentration c_i of species i is given by $c_i = \rho y_i / m_i$. The heat flux $\mathcal{Q}$ is given by Fourier's law $\mathcal{Q} = -q_0 \lambda \nabla T$, where λ is the heat conductivity. The species fluxes $\mathcal{F}_k$ are modeled by a simple Fick law:

$$\mathcal{F}_k = -q_k D_k^* \nabla y_k. \tag{24}$$

The scaling parameters q_k are the free parameters which have to be calibrated in the optimization procedure. Following Hirschfelder and Curtiss [16], the diffusion coefficients in the mixture D_k^* are given by

$$D_k^* = (1 - y_k)\left(\sum_{l\neq k} \frac{x_l}{D_{kl}^{\text{bin}}}\right)^{-1},$$

with binary diffusion coefficients D_{kl}^{bin} and mole fractions x_l.

6.1. Configuration of an ozone decomposition flame

The model problem consists of a stationary ozone decomposition flame with three chemical species, ozone O_3, oxygen molecules O_2 and atoms O, described in Becker et al. [5], where all details respect to the geometry and the mechanism can be found. In order to insure that the sum over all species mass fractions sum up to 1 and to have a consistent model, the species O_2 is erased from the set of unknown species. The initial parameters are set to $q^0 = (1, 1, 1)$, so that Fourier's and Fick's laws with conventional diffusion parameters are recovered. In Fig. 6, the resulting mass fractions of O-atoms are shown indicating the flame front.

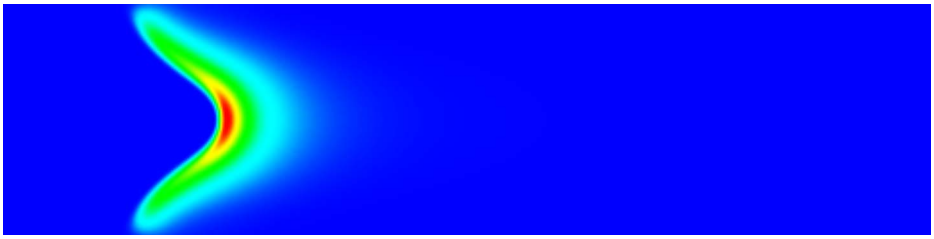

Fig. 6. Oxygen atoms for the initial diffusion model (Fick's law).

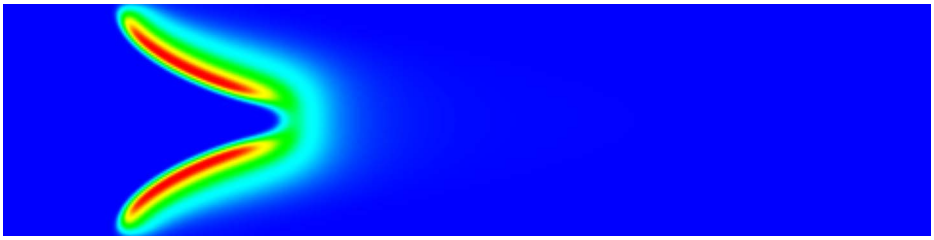

Fig. 7. Oxygen atoms of the "observations".

We substitute the experimental data by computation of the same flame but with different diffusion model. The corresponding flame is shown in Fig. 7, showing a qualitatively different flame front.

Similar to the previous example, the observation values $\overline{C} \in \mathbb{R}^{n_m}$ consist of mean values of mass fractions of oxygen atoms along $n_m = 26$ different vertical lines obtained by computations with the multicomponent diffusion models. Unlike the first example, the observation operator for the optimized parameters would not match with $\overline{C}$, because the two types of diffusion fluxes are qualitatively different. However, one may expect that optimized parameters will enhance the diffusion model at least respect to the observations.

6.2. Computational results for the ozone flame

The difference in the observation for the initial parameters q^0 is $J(u^0) = 0.09$ and a mean discrepancy

$$\bar{\varepsilon} = \frac{1}{n_m} \sum_{i=0}^{n_m} \frac{C_i - \overline{C}_i}{\overline{C}_i} = 0.3.$$

After optimization, the optimized parameters are $q_h = (0.65, 1.3828, 0.44075)$, which correspond to $J(u_h) = 0.0077$ and $\bar{\varepsilon} = 0.0062$. The comparisons of the corresponding solution, given in Fig. 8, shows nearly no difference to the observations.

With respect to the numerical algorithm, we observe that the convergence rate for the Gauß–Newton algorithm (see Section 2.1.1) is not satisfactory and that the number of iterations is too large. Therefore, we compare the Gauß–Newton algorithm (see Section 2.1.1) with the method with updates for one part of the Hessian (see Section 2.1.3). The resulting residuals of the optimization condition (7) in dependence of the number of iterations are plotted in Fig. 9. While the Gauß–Newton algorithm needs 26 iterations for reducing Res, see (23), down to 10^{-5}, only 6 iterations are needed when the matrices M_k are computed.

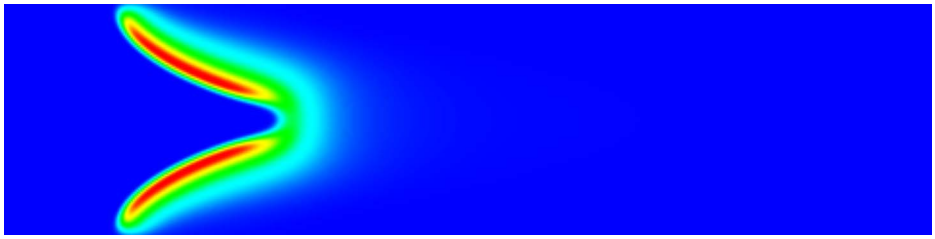

Fig. 8. Oxygen atoms for the calibrated Fick's diffusion model.

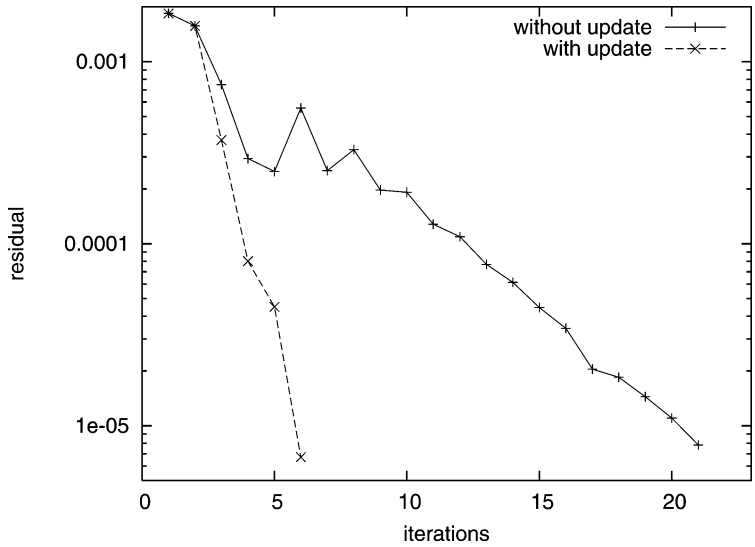

Fig. 9. Comparison of Gauß–Newton iterations and the update method of Section 2.1.3. The y-axis shows Res, the x-axis the number of iterations.

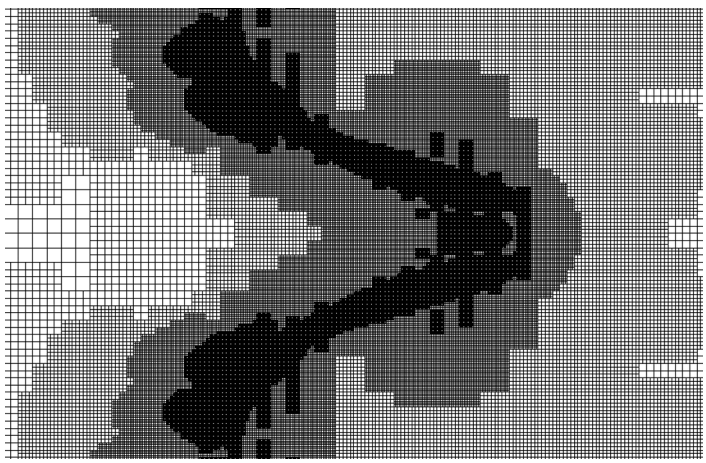

Fig. 10. A zoom of a locally refined meshes for the ozone flame.

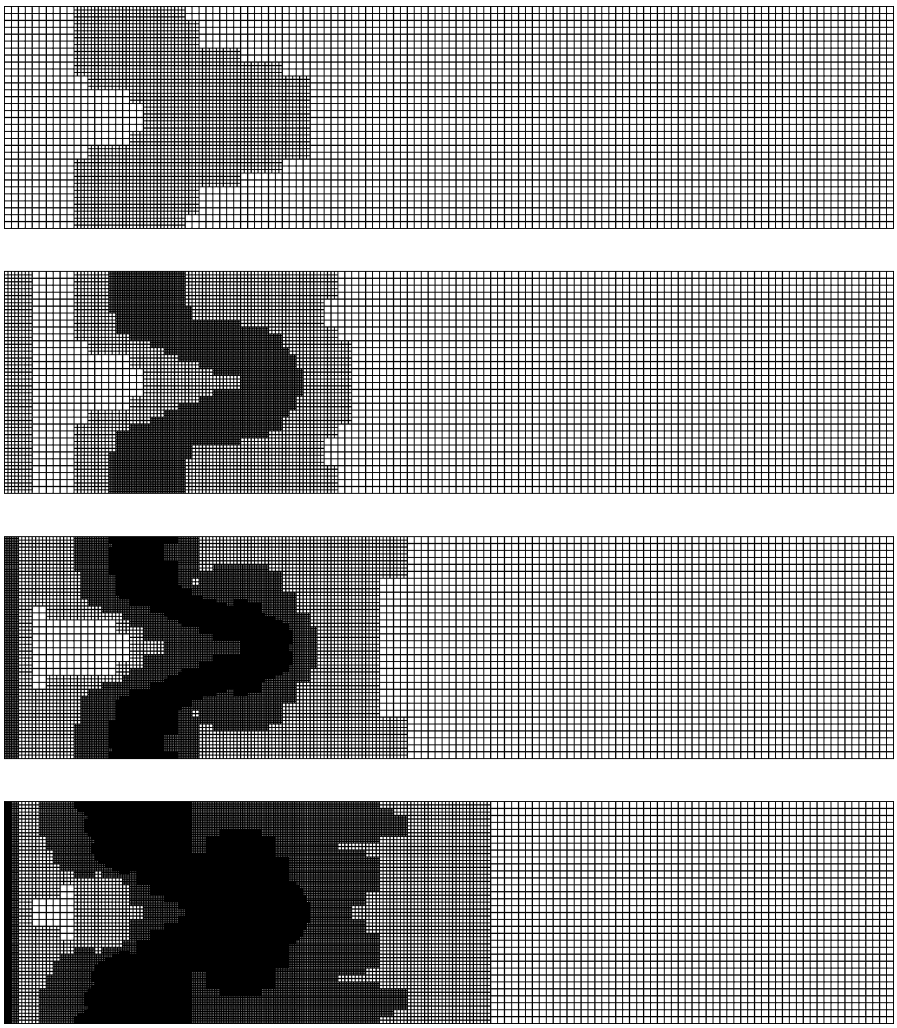

Fig. 11. A sequence of the locally refined meshes for the ozone flame.

This behavior can be explained by the fact, that even for the optimal parameters, the least-squares residual $\|C(u) - \overline{C}\|_Z$ does not vanish.

Finally, we show a zoom of the final mesh in Fig. 10 and a sequence of locally refined meshes for this optimization problem, see Fig. 11.

References

[1] R. Becker, M. Braack, A finite element pressure gradient stabilization for the Stokes equations based on local projections, Calcolo 38 (4) (2001) 173–199.

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

[3] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, in: A. Iserles (Ed.), Acta Numerica 2001, Cambridge University Press, Cambridge, 2001, pp. 1–102.
[4] R. Becker, B. Vexler, A posteriori error estimation for finite element discretization of parameter identification problems, Numer. Math. 96 (3) (2004) 435–459.
[5] R. Becker, M. Braack, R. Rannacher, Numerical simulation of laminar flames at low Mach number with adaptive finite elements, Combust. Theory Modelling 3 (1999) 503–534.
[6] M. Braack, A. Ern, A posteriori control of modeling errors and discretization errors, Multiscale Model. Simulation 1 (2) (2003) 221–238.
[7] G. Carey, J. Oden, Finite Elements, Computational Aspects, vol. III, Prentice-Hall, Englewood Cliffs, NJ, 1984.
[8] P. Ciarlet, Finite Element Methods for Elliptic Problems, North-Holland, Amsterdam, 1978.
[9] P. Clément, Approximation by finite element functions using local regularization, RAIRO Anal. Numer. 9 (1975) 77–84.
[10] A.R. Conn, N. Gould, P. Toint, Trust-Region Methods, SIAM, MPS, Philadelphia, PA, 2000.
[11] J. Dennis, R. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations equations, Classics in Applied Mathematics, SIAM.
[12] J. Dennis, D. Gay, R. Welsch, An adaptive nonlinear least-squares algorithm, ACM Trans. Math. Software 7 (1981) 348–368.
[13] J. Dieudonné, Fondation of Modern Analysis, Academic Press, New York, 1960.
[14] P. Englezos, N. Kalogerakis, Applied Parameter Estimation for Chemical Engineers, Marcel Dekker, New York, 2001.
[15] J.-L. Guermond, Stabilization of Galerkin approximations of transport equations by subgrid modeling, Modél. Math. Anal. Numér. 33 (6) (1999) 1293–1316.
[16] J.O. Hirschfelder, C.F. Curtiss, Flame and Explosion Phenomena, Williams and Wilkins, Baltimore, MD, 1949.
[17] C. Johnson, Numerical Solution of Partial Differential Equations by the Finite Element Method, Cambridge University Press, Cambridge, 1987.
[18] T.W. Lohmann, Modelling of reaction kinetics in coal pyrolysis, in: J. Warnatz, F. Behrendt (Eds.), Proceedings of the International Workshop: Modelling of Chemical Reaction Systems, 1996, http://reaflow.iwr.uni-heidelberg.de/~crs96/Program/Contrib/a28lo.htm.
[19] J. Nocedal, S. Wright, Numerical Optimization, Springer Series in Operations Research, Springer, New York, 1999.
[20] M.W. Ziesse, H.G. Bock, J.V. Gallitzendörfer, J.P. Schlöder, Parameter estimation in multispecies transport reaction systems using parallel algorithms, in: J. Gottlieb, P. DuChateaux (Eds.), Parameter Identification and Inverse Problems in Hydrology, Geology and Ecology, Kluwer Academic, Dordrecht, 1996.