INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **51**:981–999
Published online 28 December 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1160

Stabilized finite elements for 3D reactive flows

M. Braack[‡] and Th. Richter[*,†]

Institute of Applied Mathematics, University of Heidelberg, Germany

SUMMARY

Objective of this work is the numerical solution of chemically reacting flows in three dimensions described by detailed reaction mechanism. The contemplated problems include, e.g. burners with 3D geometry. Contrary to the usual operator splitting method the equations are treated fully coupled with a Newton solver. This leads to the necessity of the solution of large linear non-symmetric, indefinite systems. Due to the complexity of the regarded problems we combine a variety of numerical methods, as there are goal-oriented adaptive mesh refinement, a parallel multigrid solver for the linear systems and economical stabilization techniques for the stiff problems.

By blocking the solution components for every ansatz function and applying special matrix structures for each block of degrees of freedom, we can significantly reduce the required memory effort without worsening the convergence. Considering the Galerkin formulation of the regarded problems this is established by using lumping of the mass matrix and the chemical source terms. However, this technique is not longer feasible for 'standard' stabilized finite elements as for instance Galerkin least squares techniques or streamline diffusion. Those stabilized schemes are well established for Navier–Stokes flows but for reactive flows, they introduce many further couplings into the system compared to Galerkin formulations. In this work, we discuss this issue in connection with combustion in more detail and propose the local projection stabilization technique for reactive flows. Beside the robustness of the arising linear systems we are able to maintain the problem-adapted matrix structures presented above. Finally, we will present numerical results for the proposed methods. In particular, we simulate a methane burner with a detailed reaction system involving 15 chemical species and 84 elementary reactions.

KEY WORDS: finite elements; reactive flows; stabilization; compressible

1. INTRODUCTION

In this work, recent developments in the design and implementation of finite element methods for flow problems including chemical reactions with large heat release are described. The

*Correspondence to: Thomas Richter, INF 294, 69120 Heidelberg, Germany.
†E-mail: thomas.richter@iwr.uni-heidelberg.de
‡E-mail: malte.braack@iwr.uni-heidelberg.de

Contract/grant sponsor: SFB 359, Institute for Applied Mathematics

Received 2 June 2005
Revised 31 October 2005
Accepted 7 November 2005

emphasize is on the low-Mach number regime including the limit case of incompressible flow.

The standard Galerkin finite element method for flow problems may suffer due to the violation of the discrete inf–sup (or Babuska–Brezzi) condition for velocity and pressure approximation and, in the case of dominating advection or reaction, due to the convective terms. The streamline-upwind Petrov–Galerkin (SUPG) method, introduced by Brooks and Hughes [1], and the pressure stabilization (PSPG), introduced in References [2, 3], opened up the possibility to treat both problems in a unique framework. Additionally to the Galerkin part, the elementwise residuals are tested against appropriate test functions. This gives the possibility to use rather arbitrary finite element approximations of velocity and pressure, including equal-order pairs.

Despite the success of this classical stabilization approach to incompressible flows over the last 20 years, one can find in recent papers a critical evaluation of this approach, see, e.g. References [4, 5]. Drawbacks are basically due to the strong additional coupling between velocity and pressure in the stabilizing terms. We will show in this work, that additional couplings are even more critical for reactive flow when the convective terms of the chemical species are treated by SUPG.

For incompressible flow, new methods aim to relax the strong coupling of velocity and pressure and to introduce symmetric versions of the stabilization terms, see e.g. the global projection of Codina [6], local projection techniques (LPS) by Becker and Braack [7, 8] and Braack and Burman [9], or edge stabilization of Burman *et al.* [5] based on interior penalty techniques. In this work, we extend the LPS technique to reactive flow, described by the compressible Navier–Stokes equations with additional convection–diffusion–reaction equations for chemical species. The method is applied to combustion problems where strong heat release enforces a strong coupling between the chemical variables and flow variables. This stabilization does not affect the inter-species couplings, so that the only coupling between different species remains due to the zero-order chemical source term. This aspect will be used in the sparsity pattern of a block matrix in order to reduce the numerical costs substantially. This allows us to compute combustion problems with about 20 chemical species (and even more) in 3D using a small PC-cluster using a fully implicit scheme.

In order to illustrate a major difficulty for efficient computation of reactive flows including many chemical species we consider a single stationary convection–diffusion–reaction for species y_k:

$$\beta \cdot \nabla y_k - \operatorname{div}(D_k \nabla y_k) = f_k \tag{1}$$

with a source term $f_k = f_k(T, y_1, \ldots, y_{n_s})$ depending on the temperature and other chemical species $y_1, \ldots, y_{n_s}$. A pure Galerkin method for seeking a discrete solution, $y_{h,k}$, reads

$$\int_\Omega (\beta \cdot \nabla y_k \phi + D_k \nabla y_k \nabla \phi - f_k \phi)\, \mathrm{d}x = 0 \quad \forall \phi \in V_h \tag{2}$$

with an appropriate discrete space V_h. In the corresponding stiffness matrix, the mass fractions of different chemical species are coupled due to the zero-order term (f_k, ϕ), because $f_k = f_k(T, y_1, \ldots, y_{n_s})$. These couplings only include the degrees of freedom corresponding to the same mesh points when the mass matrix is lumped. The sparsity pattern of the stiffness matrix should take this feature into account, see Reference [10]. Now we will show that the

application of standard finite element stabilization techniques introduce further inter-species couplings which cannot be avoided by mass lumping.

In the interesting case of convection-dominated flow, the advection term $\beta \cdot \nabla y_k$ must be stabilized. Established methods are of upwind type. In the case of finite element discretization, the SUPG method is widely used because it is more accurate than simple upwinding. The principal idea is to add to the pure Galerkin formulation (2) the residual multiplied with test functions $\beta \cdot \nabla \phi$:

$$\int_\Omega (\beta \cdot \nabla y_k \phi + D_k \nabla y_k \nabla \phi - f_k \phi)\,\mathrm{d}x + \sum_{K \in \mathscr{T}_h} \tau_K \int_K (\beta \cdot \nabla y_k - \operatorname{div}(D_k \nabla y_k) - f_k)\beta \cdot \nabla \phi \,\mathrm{d}x = 0 \quad \forall \phi \in V_h$$

with element-dependent parameters τ_K. Due to these additional terms, further inter-species coupling are introduced: The product of chemical source term and SUPG test function, $f_k \beta \cdot \nabla \phi$, couple degrees of freedom from different mesh points and different chemical species. This is the reason why the SUPG stabilization enlarges the number of coupling substantially.

Thus, non-standard finite element stabilization techniques are highly relevant for reactive flow computations. In this work, we document on the use of local projection stabilization techniques in order to overcome the limitations of SUPG techniques. In addition to the robust treatment of convective terms this technique stabilizes the stiff pressure–velocity coupling for equal-order finite elements.

Furthermore, we use residual-driven *a posteriori* mesh refinement, fully coupled defect-correction iteration for linearization, and optimal multigrid preconditioning. The potential of automatic mesh adaptation together with multilevel techniques is illustrated by 3D simulations including detailed reaction mechanisms for laminar methane combustion.

We start in Section 2 with the description of the stabilization for incompressible flows without chemistry and address some aspects of implementing this method. A theoretical analysis of LPS stabilization can be found in References [7–9]. In the review article Reference [11] a comparison with residual-based and edge stabilization is given. In order to document on the order of the proposed finite element method we make a numerical comparison with a standard SUPG method.

In Section 3 we extend the discretization to reactive flows. Emphasis is given on the treatment of the preconditioner using problem-adapted sparse block matrices. The matrix structure is presented in more detail in Section 4.

Finally, we document in Section 5 on the simulation of a 3D laminar methane burner including 15 chemical species.

2. NAVIER–STOKES

In this section we will discuss stabilization techniques for the Navier–Stokes equations for velocities v and pressure p:

$$\operatorname{div} v = 0$$

$$\nu \Delta v + (v \cdot \nabla)v + \nabla p = f$$

These two variables are sampled together in the vector $u := \{p, v\}$. By $\widehat{u}$ we denote an extension of non-homogeneous Dirichlet conditions into the domain Ω.

2.1. Variational formulation for Navier–Stokes

We use the usual notation $L^2(\Omega)$ for the space of square-integrable functions in Ω, and $H^1(\Omega)$ for the Sobolev space of functions with first derivatives in $L^2(\Omega)$. The solution is sought in the space $\widehat{u} + X$, where X is a functional Hilbert space considered as a product of Hilbert spaces for each component, $L^2(\Omega) \times H^1(\Omega)^{d+1+s}$, where we denote by d the spatial dimension and by s the number of chemical species, and with standard modifications to build in the boundary conditions and probably restrictions on the mean of the pressure.

With the bilinear form given by

$$a(u)(\varphi) := \int_\Omega \operatorname{div} v\xi \, \mathrm{d}x + \int_\Omega (\nu \nabla v \nabla \phi + (v \cdot \nabla) v\phi - p \operatorname{div} \phi) \, \mathrm{d}x \tag{3}$$

together with appropriate boundary conditions, the continuous solution $u \in X$ fulfills the equation

$$a(u)(\varphi) = 0 \quad \forall \varphi \in X$$

2.2. Galerkin formulation on locally refined meshes

For the discretization we use a conforming equal-order Galerkin finite element method defined on quadrilateral (hexahedrals in 3D) meshes $\mathscr{T}_h = \{K\}$ over Ω, with elements denoted by K. The mesh parameter h is defined as a elementwise constant function by setting $h|_K = h_K$, where h_K is the diameter of K. In order to ease the mesh refinement we allow the element to have nodes, which lie on midpoints of faces of neighbouring elements. But at most one such *hanging node* is permitted for each face.

The discrete function space $V_h^{(r)}$ consist of continuous, piecewise polynomial functions of (so-called Q_r-elements) for all unknowns,

$$V_h^{(r)} = \{\varphi_h \in C(\overline{\Omega}); \varphi_h|_K \in Q_r(K) \, \forall K \in \mathscr{T}_h\} \tag{4}$$

where $Q_r(K)$ is the space of functions obtained by transformations of (iso-parametric) polynomials of order r in each spatial direction from a fixed reference unit element $\hat{K}$ to K. For a detailed description of this standard construction, see Reference [12] or [13].

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. For implementational details, see, e.g. Reference [14].

Additionally, we use patch-structured meshes. By a 'patch' of elements, we denote a group of elements (that is eight hexes in 3D), which have a common father-element in the coarser mesh $\mathscr{T}_{2h}$. Figure 1 gives an example for these patch structures for a quadrilateral and a triangular 2D mesh. Obviously, every mesh can be transformed into a patch-structured mesh by one global refinement. Alternatively, a construction of the patches by agglomeration of elements is possible in most cases.

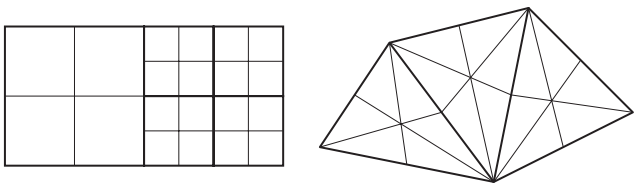

Figure 1. Quadrilateral and triangular mesh with patch structure.

There are two reasons to restrict to this patch-structured meshes:

1. Different function spaces on the same mesh can be employed. Similar to (4) we define the space of continuous, piecewise polynomial functions of the same degree but on the coarse mesh $\mathscr{T}_{2h}$,

$$V_{2h}^{(r)} := \{\varphi_h \in C(\overline{\Omega}); \varphi_h|_K \in Q_r(K)\,\forall K \in \mathscr{T}_{2h}\}$$

   This function spaces is a subset of the previous one, $V_{2h}^{(r)} \subset V_h^{(r)}$. The stabilization techniques described in the next sections will use the interpolation operator $I_{2h}^{(r)} : V_h^{(r)} \to V_{2h}^{(r)}$ into this coarser space.
2. On the patched meshes we can also establish a function space with higher polynomial degree $2r$:

$$V_{2h}^{(2r)} := \{\varphi_h \in C(\overline{\Omega}); \varphi_h|_K \in Q_{2r}(K)\,\forall K \in \mathscr{T}_{2h}\}$$

   The interpolation into this function space $I_{2h}^{(2r)} : V_h^{(r)} \to V_{2h}^{(2r)}$ is used during error estimation for local recovery. Using super-convergence arguments, we expect the difference between the solution and this higher-order interpolation to be a good approximation to the error: $\|I_{2h}^{(2r)} u_h - u_h\|_K \approx \|u - u_h\|_K$. For more details on this kind of *a posteriori* error estimation we refer to Reference [3].

Since we take for each component of the system the spaces $V_h^{(r)}$ (with standard modifications for Dirichlet conditions), the discrete space X_h is a tensor product of the spaces $V_h^{(r)}$. The discrete Galerkin solution $u_h \in \widehat{u} + X_h$ for a finite-dimensional subspace $X_h \subset X$ reads:

$$u_h \in \widehat{u} + X_h : \quad a(u_h)(\varphi) = 0 \quad \forall \varphi \in X_h \tag{5}$$

The formulation (5) is not stable in general due to the following two reasons: (i) violation of the discrete inf–sup (or Babuska–Brezzi) condition for velocity and pressure approximation and (ii) dominating advection (and reaction). Both issues will be addressed in more detail in the following.

2.3. Drawbacks of residual-based methods

The classical streamline diffusion (SUPG) stabilization for the incompressible Navier–Stokes problem, introduced by Brooks and Hughes [1], stabilizes the convective terms. Johnson and Saranen [2] presented an additional pressure stabilizing (PSPG) term in order to allow equal-order finite element approximations of velocity and pressure. Drawbacks of these techniques are basically due to the strong coupling between velocity and pressure in the stabilizing

terms and the non-trivial construction of efficient algebraic solvers. The treatment of time-dependent problems requires the usage of space–time elements. Furthermore, the numerical effort for setting up the stabilization terms is very high, mainly due to the presence of second derivatives. For the Navier–Stokes equations the stabilized form (3) with SUPG/PSPG reads

$$a_h(u)(\varphi) = a(u)(\varphi) + \sum_{K\in\mathscr{T}_h} \int_K (-\nu\Delta v + (v\cdot\nabla)v + \nabla p)(\alpha_K \nabla\xi + \delta_K (v\cdot\nabla)\phi)\,\mathrm{d}x \tag{6}$$

with elementwise constant parameters α_K and δ_K depending on the local balance of convection and diffusion. Numerical studies on the accuracy as well as the robustness of the solver will be presented later in this work.

In the next section we present an alternative stabilization technique firstly introduced for the Stokes system in Reference [7] and extended to the Oseen system in Reference [9] which circumvents most problems connected to residual-based stabilization methods.

2.4. Stabilization by local projection

The bilinear form for the Navier–Stokes equations using local projection stabilization is of the form:

$$u_h \in \widehat{u} + X_h:\ a(u_h)(\varphi) + s_h(u_h)(\varphi) = 0 \quad \forall\varphi\in X_h \tag{7}$$

with $a(u_h)(\varphi)$ defined in (3). The term $s_h(u_h)(\varphi)$ accounts for the saddle-point structure of the velocity and pressure coupling and for the convective terms.

In order to define $s_h(\cdot)(\cdot)$ we use a fluctuation operator by the difference of the identity and the previously introduced nodal interpolator $I_{2h}^{(r)}$:

$$\kappa_h : V_h \to V_h, \quad \kappa_h\phi := \phi - I_{2h}^{(r)}\phi$$

With this notation, the stabilization term added to the Galerkin formulation for an equation of type (3) is symmetric and reads

$$s_h(u_h)(\varphi) = \sum_{K\in\mathscr{T}_{2h}} \left\{ \alpha_K \int_K \nabla\kappa_h p_h\, \nabla\kappa_h\xi\,\mathrm{d}x + \delta_K \int_K (v_h\cdot\nabla)\kappa_h v_h (v_h\cdot\nabla)\kappa_h\phi\,\mathrm{d}x \right\} \tag{8}$$

where the parameters α_K, δ_K are chosen patchwise constant depending on the local balance of convection and diffusion:

$$\delta_h|_K := \frac{\delta_0 h_K^2}{6\nu + h_K\|\beta\|_{\infty,K}}$$

Here, the quantity $\|\beta\|_{\infty,K}$ is the maximum of v_h on the element K. The parameter δ_0 is a fixed constant, usually chosen as $\delta_0 = 0.5$. The elementwise parameter α_K is chosen in the same way, with some constant α_0, again usually set to $\alpha_0 = 0.5$. Note, that κ_h vanishes on V_{2h}, and therefore, the stabilization vanishes for test functions of the coarse grid $\varphi\in V_{2h}$.

For a stability proof and an error analysis for the Stokes equation we refer to Reference [7]. The proposed stabilization is consistent in the sense that the introduced terms vanish for $h\to 0$.

In contrast to the residual-based stabilization term (6) no additional couplings between pressure and the velocity are introduced. The stabilization of convection and of the pressure is completely separated. Furthermore, no second derivatives are needed to assemble the stabilization terms. Comparisons between both types of stabilization will be given later.

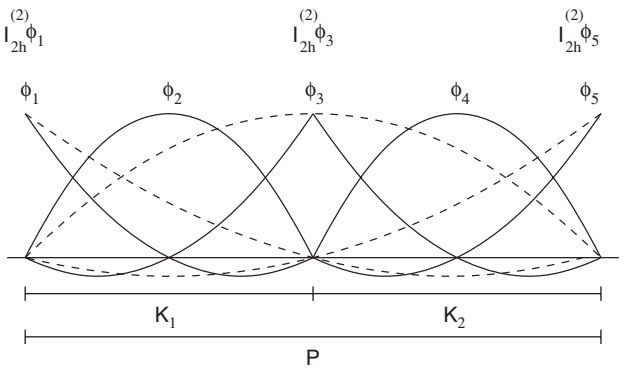

Figure 2. Patch structure of elements and nested test functions for local projection.

2.5. *Implementation of local projection stabilization for* $r=2$

The application of the projection operator is based on the nested configuration of the function spaces V_h and V_{2h} and is performed on the algebraic level with help of a local matrix vector multiplication. In order to illustrate the principles, we first discuss the 1D case. The 2D and 3D test functions are assembled as tensor products of the 1D functions.

For quadratic elements ($r=2$), the 1D test functions for a patch P and the two elements $K_1, K_2 \subset P$ are shown in Figure 2. While we have $K_i \in \mathscr{T}_h$, the patch P itself is a element in the coarser mesh $P \in \mathscr{T}_{2h}$. To assemble the stabilization term (8) on the patch P one has to arrange fluctuations, e.g. $\kappa_h p = p_h - I_{2h}^{(r)} p_h$. We express $p_h|_P$ in terms of the standard nodal basis functions $\phi_1, \ldots, \phi_5$ and corresponding nodal values:

$$p_h|_P = \langle P, \Phi \rangle := \sum_{i=1}^{5} P_i \Phi_i$$

where $P = (P_1, \ldots, P_5)$ stands for the vector of nodal values and $\Phi = (\phi_1, \ldots, \phi_5)$ for the vector of basis functions. Due to linearity, κ_h has to be applied on the test functions Φ only:

$$\kappa_h p_h|_P = \langle P, K\Phi \rangle$$

with a matrix $K \in \mathbb{R}^{5\times 5}$. Assuming the numeration of the basis functions according to Figure 2, their interpolation $I_{2h}^{(2)}$ vanishes for $i = 2, 4$:

$$I_{2h}^{(2)}\phi_2 = I_{2h}^{(2)}\phi_4 = 0$$

The interpolation of the remaining nodal functions, $I_{2h}^{(2)}\phi_j$, $j = 1, 3, 5$, can be represented as a matrix multiplied with the fine grid basis:

$$\begin{pmatrix} I_{2h}^{(2)}\phi_1 \\ I_{2h}^{(2)}\phi_3 \\ I_{2h}^{(2)}\phi_5 \end{pmatrix} = \begin{pmatrix} 1 & 3/8 & 0 & -1/8 & 0 \\ 0 & 3/4 & 1 & 3/4 & 0 \\ 0 & -1/8 & 0 & 3/8 & 1 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix}$$

Thus, on a patch P, the matrix K is given by

$$K = \begin{pmatrix} 0 & -\frac{3}{8} & 0 & \frac{1}{8} & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -\frac{3}{4} & 0 & -\frac{3}{4} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -\frac{1}{8} & 0 & -\frac{3}{8} & 0 \end{pmatrix}$$

Since the projection is always performed on the reference element, this matrix only depends on the degree of the basis functions. A similar matrix can be constructed for linear finite element spaces or for higher-order spaces.

For the transition to higher spatial dimensions, we assemble the test functions as tensor products of the 1D functions ϕ_i. In two dimension, with

$$\phi_{i,j}(x, y) := \phi_i(x)\phi_j(y)$$

and the vector $\Phi^{2D} = (\phi_{i,j})$, we get the following representation of the fluctuation operator:

$$(K^{2D}\Phi^{2D})_{ij} = (K\Phi)_i(K\Phi)_j = \sum_{k,l} \Pi_{i,k}\Pi_{j,l}\phi_{k,l}$$

with $K^{2D} \in \mathbb{R}^{5^2 \times 5^2}$.

Let us consider the pressure stabilization term $\int_K \nabla\kappa_h p_h \nabla\kappa_h \phi$ in (8) with a basis function $\phi = \phi_i$. This term turns out to be

$$\int_K \nabla\kappa_h p_h \nabla\kappa_h \phi_i = \int_K \langle P, \nabla K^{2D}\Phi\rangle(\nabla K^{2D}\Phi)_i \, dx$$

We recapitulate that K^{2D} is the fixed matrix given above, P is the vector of nodal values of the discrete pressure p_h on the patch K, and Φ is the vector of nodal basis functions on P. The other terms in (8) are obtained analogously.

The application of the projection operator enlarges the matrix stencil due to the interpolation into the coarse function space V_{2h}. In three spatial dimensions using triquadratic finite elements, one patch includes 2744 matrix couplings instead of 512 couplings necessary for the Galerkin part in one element. However, it is possible to use another projection operator in the matrix of the linear system. This slightly reduces the Newton convergence but substantially minimizes the memory usage. Within the matrix we use as projection operator the fluctuations with regard to a polynomial space of lower degree on the same mesh:

$$\tilde{\kappa}_h p_h := p_h - I_h^{(1)} p_h$$

This projection forms another local projection method itself. Although this method is favourable in terms of effort and memory usage, it cannot be recommended in general for computing the residuals, since the accuracy of the resulting scheme is reduced by one order. Here, we use this projection only as a cheap preconditioner.

Actually, the local projection method is a large set of techniques. The definition of the operator κ_h via the interpolation to the coarse space $V_{2h}^{(2)}$ is one possibility. Another type of local projection is the stabilization term

$$\alpha_K \int_K (\nabla p_h - \overline{\nabla p_h}) \nabla \xi \, \mathrm{d}x$$

where $\overline{\nabla p_h}$ is a local projection of the pressure gradient to a polynomial of order $r-1$ onto a patch $P \in \mathscr{T}_{2h}$. The stabilization of the convective term is analogous.

2.6. Numerical study for Navier–Stokes

In this section we perform a numerical study in 2D to examine the accuracy of the local projection stabilization compared to the SUPG/PSPG method. As a test case we use the well-established benchmark problem 'Flow around a cylinder' described in Reference [15]. A circular obstacle is embedded into a channel. Quantity of interest is the drag coefficient on this obstacle in the flow domain. On Γ_{in} a parabolic inflow profile is given with a maximum velocity of 0.3 m/s, which yields the Reynolds number $Re = 20$. The value of interest is the drag defined by

$$c_{\text{D}} = C \int_S \left(\nu \frac{\partial v_t}{\partial n} n_y - p n_x \right) \mathrm{d}s$$

with a constant C. Instead of using this boundary integral for the evaluation of the drag, the integral can be transformed into an integral over the whole domain (see Reference [16]) for details. Using this alternative evaluation method, the accuracy is enhanced to $O(h^4)$ for biquadratic elements.

In Figure 3 the error for the local projection stabilization is plotted and compared with SUPG/PSPG. The parameters α_0 and δ_0 are chosen as 0.5 in both cases. Considering this

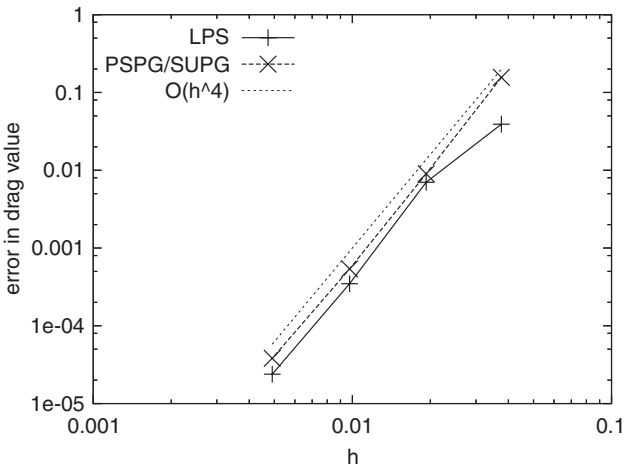

Figure 3. Accuracy of the drag evaluation on a sequence of globally refined meshes. The stabilization parameters α_0 and δ_0 are chosen as 0.5. The solid line belongs to the local projection method, the dashed line to the residual method, the dotted line illustrates fourth order convergence.

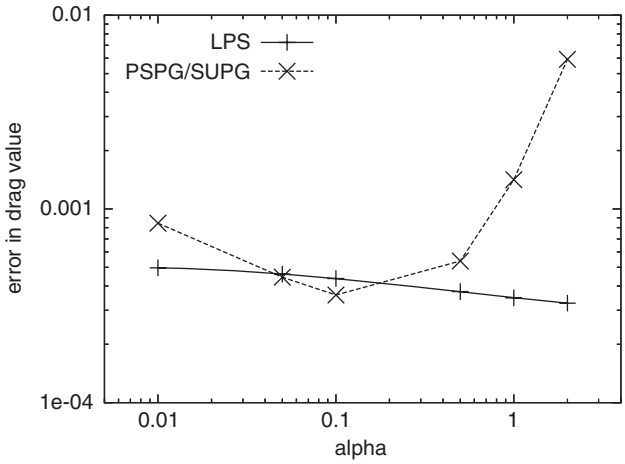

Figure 4. Accuracy of the drag evaluation on a mesh with 10 240 elements with regard to the choice of the stabilization parameter α_0. δ_0 is chosen as 0.5. The solid line belongs to the local projection method, the dashed line to the residual method.

Table I. Mean convergence rate of multigrid solver with biquadratic finite elements. Upper table: local projection stabilization; lower table: PSPG/SUPG. Missing numbers indicate divergence of the linear solver.

Elements	$\delta = 0.01$	$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.5$	$\delta = 1$	$\delta = 2$
LPS						
2560	0.125	0.063	0.071	0.112	0.152	0.260
10 240	0.503	0.062	0.066	0.102	0.128	0.224
PSPG/SUPG						
2560	0.356	0.243	0.227	0.357	0.604	—
10 240	0.761	0.250	0.204	0.461	—	—

'good' choice of the parameters—which generally depends on the specific problem—both stabilization techniques result in a comparable accuracy.

In Figure 4 the error on two sequenced meshes with regard to the choice of the parameter α_0 is shown. While the local projection method produces good results for the drag coefficient on all choices of α_0, the residual method heavily depends on the correct choice. Regarding the accuracy, the influence of the parameter δ_0 is negligible for both settings. This is due to the low Reynolds number in this benchmark problem.

However, even more dramatic than the dependence of the accuracy on α_0 is the robustness of the linear solver w.r.t. the choice of the parameters. All linear problems are solved with a geometric multigrid solver based on 'global coarsening', see Reference [17] or [18] for details. In Table I we list convergency rates of this multigrid solver. For 'wrong' choices of δ_0 SUPG/PSPG stabilization leads to a breakdown of the linear solver.

3. REACTIVE FLOW PROBLEMS

3.1. Governing equations for reactive flow problems

As before, we denote the velocity by v and the pressure by p. Additionally, we have the temperature T and the density ρ. Furthermore, we have n_s species mass fractions denoted by y_k, $k=1,\ldots,n_s$. The basic equations for reactive viscous flow express the conservation of total mass, momentum, energy, and species mass in the following form:

$$\partial_t \rho + \operatorname{div}(\rho v) = 0 \tag{9}$$

$$\rho \partial_t v + \rho(v \cdot \nabla)v - \operatorname{div} \pi + \nabla p = g\rho \tag{10}$$

$$\rho c_p \partial_t T + \rho c_p v \cdot \nabla T - \operatorname{div} \lambda \nabla T = -\sum_{k=1}^{n_s} h_k m_k \dot{\omega}_k \tag{11}$$

$$\rho \partial_t y_k + \rho v \cdot \nabla y_k + \operatorname{div} \mathscr{F}_k = m_k \dot{\omega}_k \quad k=1,\ldots,n_s \tag{12}$$

where g is the gravitational force, c_p is the heat capacity of the mixture at constant pressure, and for each species k, m_k is its molar weight, h_k its specific enthalpy, $\dot{\omega}_k$ its molar production rate. We consider Fick's law for the species mass diffusion fluxes $\mathscr{F}_k$ driven by gradients of mass fractions, see Reference [19]:

$$\mathscr{F}_k = -\rho D_k^* \nabla y_k, \quad k=1,\ldots,n_s \tag{13}$$

The viscous stress tensor π is given by

$$\pi = \mu(\nabla v + (\nabla v)^T - \tfrac{2}{3} \operatorname{div} v I)$$

The diffusion coefficients D_k^*, the thermal conductivity λ and the viscosity μ are functions of y and T. The conservation equations (9)–(12) are completed by the ideal gas law for mixtures

$$\rho = \frac{p\overline{m}}{RT} \tag{14}$$

with the universal gas constant $R = 8.31451$, and mean molar weight $\overline{m}$ given by

$$\overline{m} = \left(\sum_{k=1}^{n_s} \frac{y_k}{m_k} \right)^{-1}$$

For more details on the derivation of the equations and the chemical reactions, see Reference [20].

Equations (9)–(12) are linearly dependent because the species mass fractions sum up to unity and

$$\sum_{k=1}^{n_s} \mathscr{F}_k = \sum_{k=1}^{n_s} m_k \dot{\omega}_k = 0$$

Therefore, we omit one equation in (12), say that of the last species, and set with $s := n_s - 1$,

$$y_{n_s} := 1 - \sum_{i=1}^{s} y_i$$

The system of equations is closed by suitable boundary conditions depending on the specific configuration to be considered: for temperature and species, we allow for non-homogeneous

Dirichlet and homogeneous Neumann conditions. For the velocity v, we allow for non-homogeneous Dirichlet conditions at the inflow and rigid walls, and for the natural outflow boundary condition.

In order to account for compressible flows at low Mach number, the total pressure is split in two parts

$$p(x,t) = p_{\text{th}}(t) + p_{\text{hyd}}(x,t)$$

While the so-called thermodynamic pressure $p_{\text{th}}(t)$ is constant in space, the hydrodynamic pressure part $p_{\text{hyd}}(x,t)$ may vary in space and time. Hence, the pressure gradient in the momentum equation (10) can be replaced by ∇p_{hyd}. This is important for flows at low Mach number, where p_{hyd} is smaller than p_{th} by several magnitudes.

The vector u assembles the variables $u := \{p_{\text{hyd}}, v, T, y_1, \ldots, y_s\}$ while the density is considered as a coefficient determined by the ideal gas law

$$\rho = \frac{(p_{\text{th}} + p_{\text{hyd}})\overline{m}}{RT} \tag{15}$$

3.2. *Stabilized finite elements for reactive flow*

We define the semilinear form for stationary solutions of (9)–(14):

$$\begin{aligned} a(u)(\varphi) := & \int_\Omega \operatorname{div}(\rho v)\xi \,\mathrm{d}x + \int_\Omega (\rho(v\cdot\nabla)v\phi + \pi\nabla\phi - p_{\text{hyd}}\operatorname{div}\phi - g\rho\phi)\,\mathrm{d}x \\ & + \int_\Omega \left(\rho c_p v\cdot\nabla T\sigma + \lambda\nabla T\,\nabla\sigma + \sum_{k=1}^{n_s} h_k m_k \dot\omega_k \sigma\right)\mathrm{d}x \\ & + \sum_{k=1}^{s}\int_\Omega (\rho v\cdot\nabla y_k\tau_k - \mathscr{F}_k\nabla\tau_k - m_k\dot\omega_k\,\tau_k)\,\mathrm{d}x \end{aligned}$$

for test functions $\varphi = \{\xi, \phi, \sigma, \tau_1, \ldots, \tau_s\}$.

As previously discussed for Navier–Stokes, this semilinear form is not stable for equal-order finite elements. Considering the full set of reactive flow equations further difficulties occur due to the additional convective terms. As already mentioned in the introduction, SUPG applied to reactive flows will introduce strong couplings between different chemical species.

The expansion of the local projection method to reactive flows is straightforward. For the full reactive flow system the stabilization term (8) consists of the previously introduced stabilization for pressure–velocity and convection stabilization for all convective terms:

$$s_h(u_h)(\varphi) = \sum_{K\in\mathscr{T}_{2h}}\int_K \left[s^p(u_h)(\xi) + s^v(u_h)(\phi) + s^T(u_j)(\psi) + \sum_{k=1}^{n_s} s^k(u_h)(\psi_k)\right]\mathrm{d}x$$

$$s^p(u)(\xi) = \alpha_K(\nabla\kappa_h p)(\nabla\kappa_h\xi)$$

$$s^v(u)(\phi) = \delta_K((\rho v\cdot\nabla)\kappa_h v)((\rho v\cdot\nabla)\kappa_h\phi)$$

$$s^T(u)(\psi) = \tau_K(\rho v\cdot\nabla\kappa_h T)(\rho v\cdot\nabla\kappa_h\psi)$$

$$s^k(u)(\psi_k) = \tau_K^k(\rho v\cdot\nabla\kappa_h y_k)(\rho v\cdot\nabla\kappa_h\psi_k)$$

The stabilization parameters τ_K, τ_K^k depend again on the local balance of convection and diffusion of the temperature and the species, respectively.

Let us shortly compare the further couplings introduced by this technique. On the one hand, the stencil becomes larger due to the projection onto patches. On the other hand, and this is the crucial point for reactive flows, the stabilization does not act on the reactive source terms $m_k\dot{\omega}_k$. Hence, no further couplings between different chemical species are introduced. We come back to this point when we discuss the matrix structure of the corresponding linear systems in a later section.

4. SOLUTION PROCEDURE

The discrete equation system (7) are solved by quasi-Newton iteration with an approximate Jacobian $J = (J_{ij})$ of the stiffness matrix with block entries,

$$J_{ij} \approx a'(u^n)(\varphi_j, \varphi_i)$$

of size $n = s + d + 2$. The corresponding linear systems are solved with a multigrid algorithm. Due to the blocking of all the components of the system an incomplete block-LU factorization can be applied. This accounts for the strong coupling of hydrodynamical variables as pressure, velocity and temperature with the chemical variables. This linear solver is described in detail in Reference [18]. Here we focus on the matrix structures because they may become extremely expensive for large chemical mechanisms.

Let us discuss the memory effort for storing a Jacobian with blocks J_{ij} of type

$$\begin{bmatrix} A_{pp} & A_{pv} & A_{pT} & A_{py} \\ A_{vp} & A_{vv} & A_{vT} & A_{vy} \\ A_{Tp} & A_{Tv} & A_{TT} & A_{Ty} \\ A_{yp} & A_{yv} & A_{yT} & A_{yy} \end{bmatrix} \tag{16}$$

Note, that the computational effort is aligned to the number of matrix entries. As already mentioned in Section 1, the biggest part in the blocks J_{ij} is due to the species couplings A_{yy}, at least for a large number of species $s \gg 1$. Considering trilinear finite elements (with the 27-point stencil on tensor grids), the matrix only containing matrix blocks of this type would have

$$27(5+s)^2$$

entries per grid point. In Table II we show the memory necessary for storing one system matrix in 2D and 3D when a reaction mechanism with $s = 15$ species is used.

The species couplings A_{yy} are due to various terms:

- For combustion problems, the chemical source terms $\dot{\omega}_k$ usually enforce extremely strong couplings between different species. However, if the mass matrix in the finite element discretization is lumped, these coupling do not appear in off-diagonal blocks J_{ij}, $i \neq j$.
- Some diffusion laws, as for instance the extended Fick's law or multicomponent diffusion, see Reference [21], show off-diagonal couplings when mass fractions y_k are used as

Table II. Memory needed for storing one system matrix in single precision and $s = 15$ species in two and three dimensions.

	2D	3D
Matrix couplings	9	27
Block size	1849	1936
Elements necessary for $\approx 5\%$ error	10 000	250 000
Memory (single prec.)	634 MB	50 GB

primary variables. However, the off-diagonal couplings due to the extended Fick's law are usually of minor importance so that they may be neglected in the Jacobian. In this case, the contribution due to diffusion are diagonal in the blocks J_{ij}. For Fick's diffusion law (13) this is the case without modification of the Jacobian.

- SUPG stabilization generate off-diagonal couplings between species gradients due to the consistence terms, see discussion in Section 2.3.

In summary, due to the local projection stabilization the matrix blocks can be classified into two types: into dense diagonal blocks J_{ii} and into sparse off-diagonal blocks J_{ij}, $i \neq j$. The diagonal blocks remain of the type (16) while the off-diagonal blocks J_{ij}, $i \neq j$ of the system matrix are stored in the form

$$J_{ij} = \begin{bmatrix} A_{pp} & A_{pv} & A_{pT} & \\ A_{vp} & A_{vv} & A_{vT} & \\ A_{Tp} & A_{Tv} & A_{TT} & \\ & & & D_{yy} \end{bmatrix}$$

with a diagonal matrix D_{yy}. Such a block has only $(5^2 + s)$ entries for three velocity components (3D) compared to $(5 + s)^2$ entries of a full block J_{ii}. Using different matrix blocks for the off-diagonals, the storage usage reduces to

$$(5 + s)^2 + 26(5^2 + s)$$

If we use a reaction mechanism with $s = 15$ species, the memory needed to store a matrix is a factor

$$[27(5 + s)^2] : [(5 + s)^2 + 26(5^2 + s)] \approx 7.5$$

smaller than using the standard sparse matrix. Note, that the saving grows with larger reaction systems. While the Newton residual is kept untouched, the Newton convergence may become slightly reduced. For these calculations we have assumed the usage of linear finite elements. However using quadratic finite elements, the proportions keep unchanged. For more details we refer to Reference [10].

5. SIMULATION OF A 3D BURNER

In this section, we apply the proposed methods to the simulation of a 3D methane burner with trilinear finite elements. The household burner constructed by BOSCH is an example of

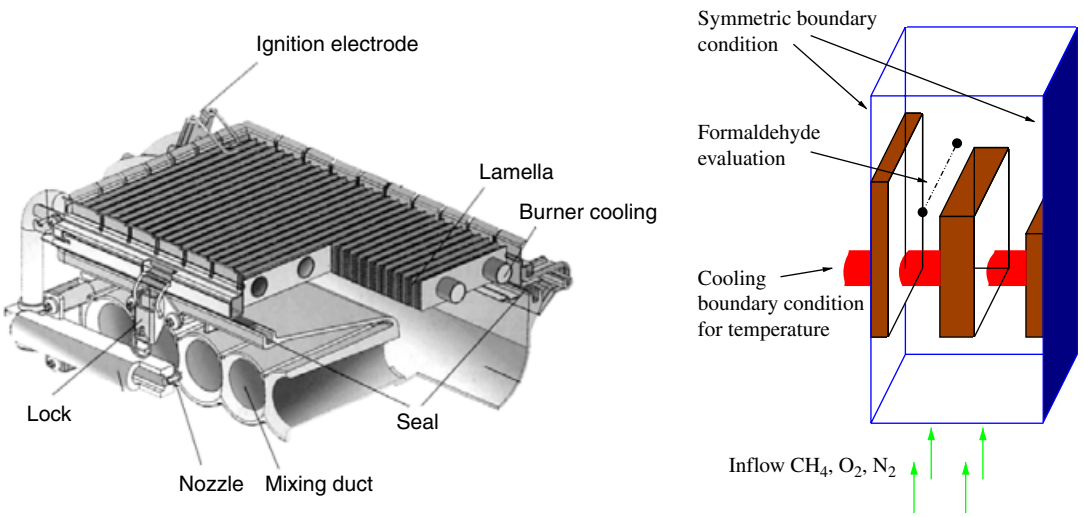

Figure 5. Left: sketch of a household burner. Right: computational domain and considered functional for 3D simulation.

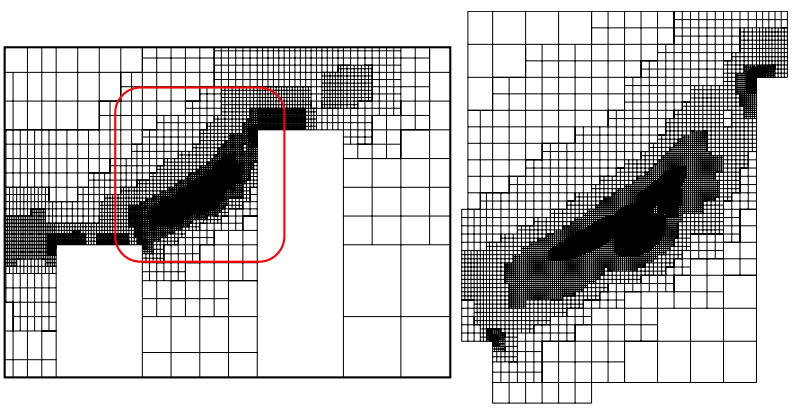

Figure 6. Cut-outs of mesh used to generate the 2D simplification for the household burner.

a burning facility with a 3D laminar stationary flame, see Figure 5. This burner consists of several slots, but in contrast to many configuration, the symmetry is violated due to several cooling ducts transversal to the lamella. In Reference [22], a 2D approximation was arranged by neglecting the cooling ducts. With the software GASCOIGNE [23] a parameter study was carried out in order to obtain information concerning pollution formation under several loads of the burner, see Reference [22]. However, the influence of the cooling ducts could not be analysed. In this section we perform simulations of a full 3D domain including the cooling devices.

As a first step towards 3D combustion simulations we use the C1 reaction mechanism with 15 chemical species (see Reference [24]). Due to the large number of chemical species, the

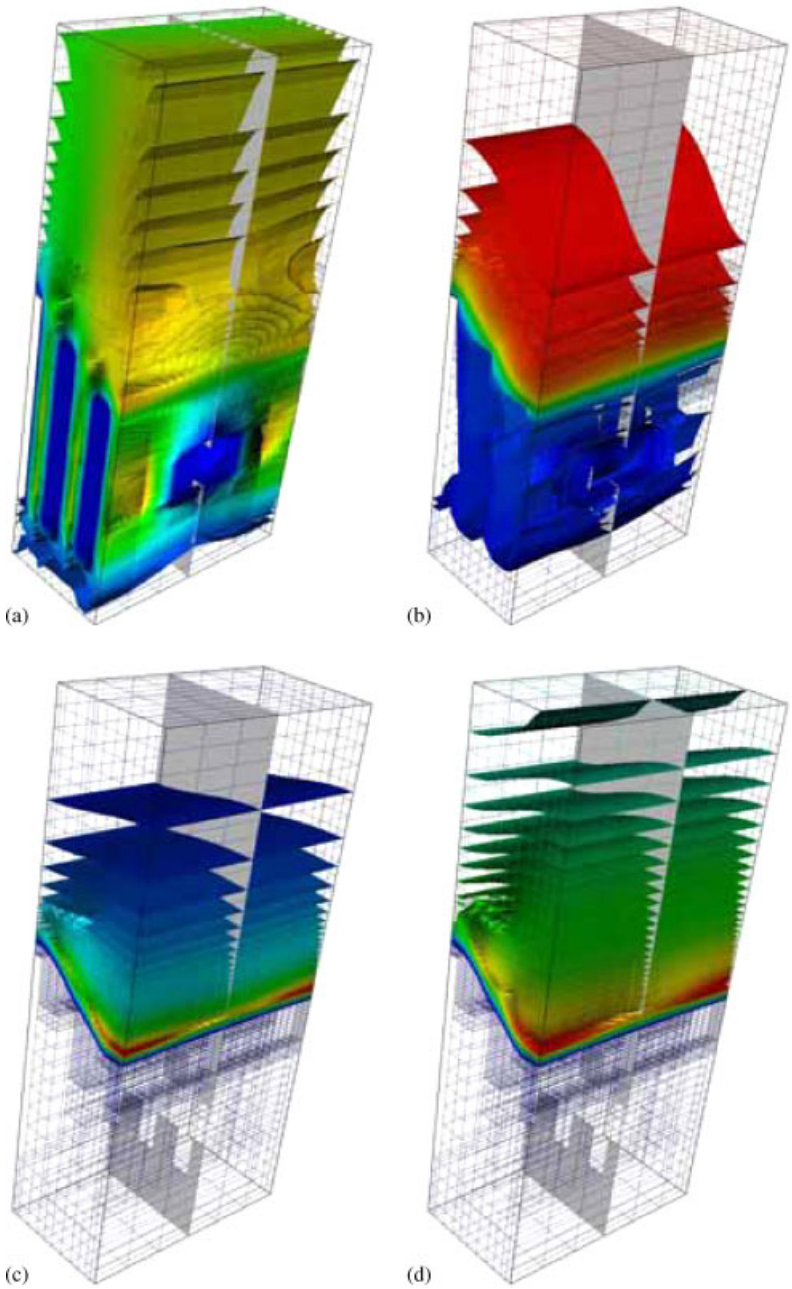

Figure 7. Simulation of the 3D methane burner: (a) vertical velocity; (b) temperature; (c) H mass fractions; and (d) OH mass fractions.

non-linearities due to chemical kinetics, the stiffness due to the differences in time scales for the fluid and the chemistry, this problem is much more complex than the corresponding flow problem without chemistry.

The sheer size of the problem with lots of chemical species leads to huge matrices, which already in terms of memory usage make the use of parallel computers inevitable. In addition, adaptive mesh refinement is used to further reduce the problem dimension.

5.1. Set-up of numerical study

The 3D simulation where initiated with a prolongation of a 2D simplification. In Figure 6, cut-outs of adaptively refined meshes from this 2D simplification are given. The mesh adaptation is driven by the dual weighted residual method (DWR) [25]. The meshes are optimized with regard to the evaluation of the formaldehyde CH_2O concentration along a line taking course in the 3D domain. In the 2D simplifications this quantity is presented by the evaluation of a single point. This evaluation point is highly resolved by the adapted meshes. Further, the edges of the lamellae are locally refined due to the produced singularities in the solution.

In Figure 5 (right) the computational domain including the cooling duct is given. The 3D simulations are performed on a PC-cluster. Details on the parallel multigrid solver are given in the PhD project of Richter [26]. As mentioned before, mesh adaption is based on estimating the line functional. Although acting as an obstacle to the flow, the cooling duct does not require relevant mesh adaption, since its boundary is smooth and no chemical reaction takes place in this area. However, the influence of the cooling may not be neglected as we will see in the following.

For obtaining a stationary flame in this case, a pseudo-time stepping with 140 iterations was necessary. The computation on a mesh with 30 928 nodes and 26 680 elements needs 6.5 hours on an Athlon cluster (1.4 GHz) with 30 processors.

5.2. Short comparison of the 2D and 3D solutions

In Figure 7, the 3D effect due to the cooling tubes are clearly visible. In particular, the flow velocity is reduced beyond the cooling tubes and the flame front becomes less pronounced.

All regarded functionals feature a significant variety in z-direction, and also an overall discrepancy in comparison to the 2D simplification is observed. In Table III we list the maximal values of these four components identified in the whole domain for the 2D and the 3D setting. The difference in the temperature of about 100 K is quite remarkable and aroused by the cooling device. The lower temperature in the 3D configuration leads to large differences in all other components (Figure 8).

Table III. Maximal values for the velocity, the temperature as well as the mass fraction of formaldehyde and HO_2 obtained in the 2D and 3D simulation.

	2D	3D
Velocity	3.39 m/s	2.61 m/s
Temperature	2156 K	2039 K
CH_2O	$7.1e^{-3}$	$5.2e^{-3}$
HO_2	$2.3e^{-4}$	$6.2e^{-4}$

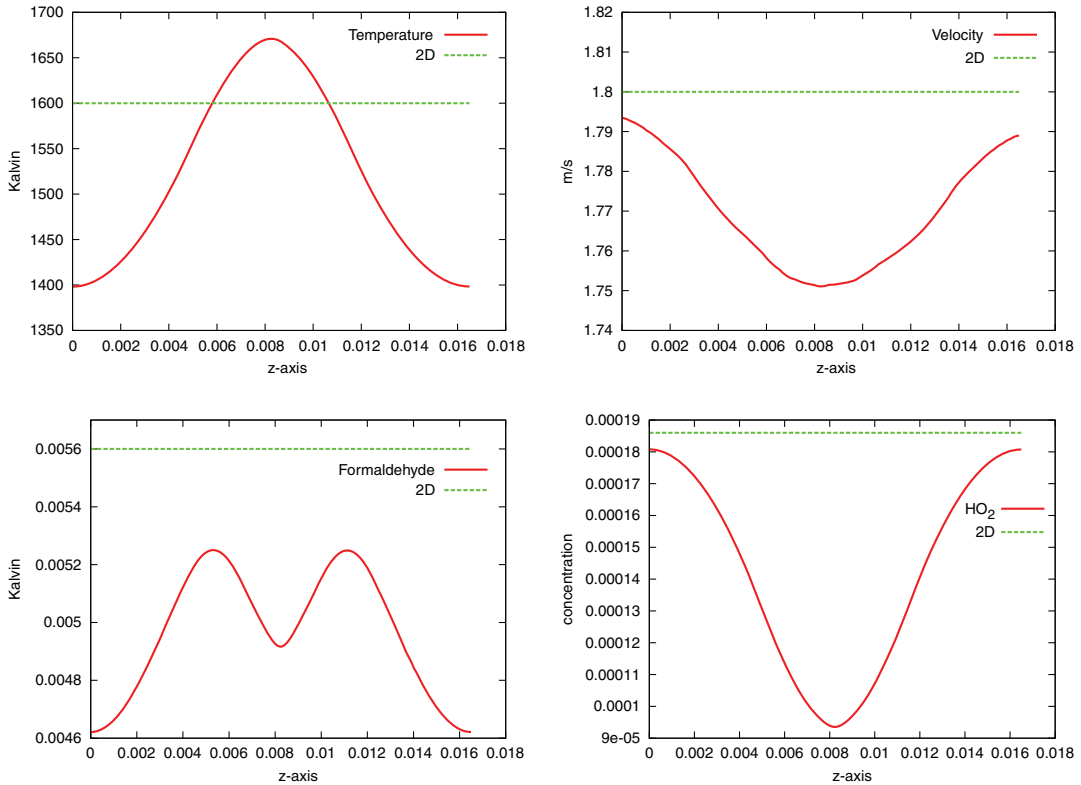

Figure 8. Comparison of 2D and 3D simulations of the household burner. Cross sections of profiles of temperature, the velocity in main flow direction, the mass fractions of formaldehyde, and HO_2-radicals along the z-axis.

Thus, recapitulating the results, the considered configuration yields real 3D features. Two-dimensional simplifications of the geometry are not justifiable. However, detailed 3D simulations are possible, if one combines efficient solvers with mesh adaptation and parallelization. We refer to Reference [26] for more details on the configuration and a comparison between the 2D and 3D simulations.

REFERENCES

1. Brooks A, Hughes T. Streamline upwind Petrov–Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
2. Johnson C, Saranen J. Streamline diffusion methods for the incompressible Euler and Navier–Stokes equations. *Mathematics of Computation* 1986; **47**:1–18.
3. Braack M, Richter T. Mesh and model adaptivity for flow problems. In *Reactive Flows, Diffusion and Transport*, Rannacher R *et al.* (eds). Springer: Berlin, 2005.
4. Gelhard T, Lube G, Olshanskii M, Starcke J. Stabilized finite element schemes with LBB-stable elements for incompressible flows. *Journal of Computational and Applied Mathematics* 2005; **177**:243–267.
5. Burman E, Fernandez M, Hansbo P. Edge stabilization for the Navier–Stokes equations: a conforming interior penalty finite element method, in preparation, 2004.

6. Codina R. Stabilization of incompressibility and convection through orthogonal subscales in finite element methods. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**(13/14):1579–1599.
7. Becker R, Braack M. A finite element pressure gradient stabilization for the Stokes equations based on local projections. *Calcolo* 2001; **38**(4):173–199.
8. Becker R, Braack M. A two-level stabilization scheme for the Navier–Stokes equations. In *Numerical Mathematics and Advanced Applications*, *ENUMATH 2003*, Feistauer M (ed.). Springer: Berlin, 2004; 123–130.
9. Braack M, Burman E. Local projection stabilization for the Oseen problem and its interpretation as a variational multiscale method. *SIAM Journal on Numerical Analysis* 2005, accepted.
10. Braack M. An adaptive finite element method for reactive flow problems. *Ph.D. Dissertation*, Universität Heidelberg, 1998.
11. Braack VJM, Burman E, Lube G. Stabilized finite element methods for the generalized Oseen problem, 2005, in preparation.
12. Ciarlet P. *Finite Element Methods for Elliptic Problems*. North-Holland: Amsterdam, 1978.
13. Johnson C. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press: Cambridge, U.K., 1987.
14. Carey G, Oden J. *Finite Elements*, *Computational Aspects*, vol. III. Prentice-Hall: Englewood Cliffs, NJ, 1984.
15. Schäfer M, Turek S. Benchmark computations of laminar flow around a cylinder (with support by Durst F, Krause E, Rannacher R). In *Flow Simulation with High-Performance Computers II. DFG Priority Research Program Results 1993–1995*, Hirschel E (ed.), Notes on Numerical Fluid Mechanics, vol. 52. Vieweg: Wiesbaden, 1996; 547–566.
16. Braack M, Richter T. Solutions of 3D Navier–Stokes benchmark problems with adaptive finite elements. *Computers and Fluids* 2006; **35**(4):372–392.
17. Becker R, Braack M. Multigrid techniques for finite elements on locally refined meshes. *Numerical Linear Algebra with Applications* 2000; **7**:363–379 (special issue).
18. Becker R, Braack M, Richter T. Parallel multigrid on locally refined meshes. In *Reactive Flows*, *Diffusion and Transport*, Rannacher R *et al.* (eds). Springer: Berlin, 2005.
19. Hirschfelder JO, Curtiss CF. *Flame and Explosion Phenomena*. Williams and Wilkins Cp: Baltimore, MD, 1949.
20. Williams FA. *Combustion Theory*. Addison-Wesley: Reading, MA, 1985.
21. Ern A, Giovangigli V. *Multicomponent Transport Algorithms*. Lecture Notes in Physics, vol. m24. Springer: Berlin, 1994.
22. Parmentier S, Braack M, Riedel U, Warnatz J. Modeling of combustion in a lamella burner. *Combustion Science and Technology* 2003; **175**(1):173–199.
23. Gascoigne3D. A high performance adaptive finite element toolkit. http://www.gascoigne.de
24. Smooke MD. Numerical modeling of laminar diffusion flames. In *Progress in Astronautics and Aeronautics*, Oran ES, Boris JP (eds), vol. 135, 1991.
25. Becker R, Rannacher R. A feed-back approach to error control in finite element methods: basic analysis and examples. *East-West Journal of Numerical Mathematics* 1996; **4**:237–264.
26. Richter T. Parallel multigrid for adaptive finite elements and its application to 3d flow problem. *Ph.D. Dissertation*, Universität Heidelberg 2005.

