STABILIZED ADAPTIVE FINITE ELEMENTS FOR LAMINAR BURNERS IN 3-D

M. Braack* and T. Richter*

*Institut für Angewandte Mathematik, Universität Heidelberg
INF 294, 69124 Heidelberg, Germany
e-mail: malte.braack@iwr.uni-heidelberg.de ,
         thomas.richter@iwr.uni-heidelberg.de

Key words: reactive flows, adaptivity, finite elements, combustion

Abstract. We describe recent developments in the design and implementation of finite element methods for the compressible Navier-Stokes equations modeling chemically reactive flows. The emphasize is on the low-Mach number regime including the limit case of incompressible flow. The most important ingredients are appropriate finite element discretizations, 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 2D and 3D simulations of laminar methane combustion including detailed reaction mechanisms.

1 UNDERLYING SET OF EQUATIONS

We denote the velocity by $v$, the pressure by $p$, the temperature by $T$ and the density by $\rho$. 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 form

$$\partial_t \rho + \text{div} (\rho v) = 0,$$

$$\rho \partial_t v + \rho (v \cdot \nabla)v - \text{div} \pi + \nabla p = g \rho,$$

$$\rho c_p \partial_t T + (\rho c_p v + \alpha) \cdot \nabla T + \text{div} Q = - \sum_{k=1}^{n_s} h_k m_k \dot{\omega}_k,$$

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

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, and $\mathcal{F}_k$ its mass diffusion flux. The expression $Q$ denotes the temperature flux. The viscous stress tensor $\pi$ is given by

$$\pi = \mu \left( \nabla v + (\nabla v)^T - \frac{2}{3} \text{div} v I \right).$$
The vector
\[ \alpha = \sum_{k=1}^{n_s} c_{pk} F_k \]  
(5)
is associated with an enthalpy flux due to diffusion fluxes of species having different specific heat capacities \(c_{pk}\). In equation (3), we have neglected heat release by pressure convection, volume viscosity effects, and viscous dissipation, because of their small impact on laminar flames. The conservation equations (1)–(4) are completed by the ideal gas law for mixtures
\[ \rho = \frac{\overline{m} T}{RT}, \]  
(6)
with the universal gas constant \(R = 8.31451\), and mean molar weight \(\overline{m}\) given by
\[ \overline{m} = \left( \sum_{k=1}^{n_s} \frac{y_k m_k}{m} \right)^{-1}. \]

We consider Fick’s law for the species mass diffusion fluxes \(F_k\) driven by gradients of mass fractions:
\[ F_F^k = -\rho D_k^* \nabla y_k, \quad k = 1, \ldots, n_s. \]  
(7)
Following Hirschfelder & Curtiss[14], the diffusion coefficients \(D_k^*\) are given by
\[ D_k^* = \frac{1 - y_k}{\sum_{l \neq k} \frac{x_l}{D_{kl}^\text{bin}}}, \]
where \(D_{kl}^\text{bin}\) is the binary diffusion coefficient for the species pair \((k, l)\). A more accurate diffusion model is the “extended Fick’s law”:
\[ F_F^k = -\rho D_k^* \frac{y_k}{x_k} \nabla x_k, \quad k = 1, \ldots, n_s. \]  
(8)
driven by gradients of mole fractions, \(x_k = \frac{\overline{m} y_k}{m_k}\). However, this diffusion model is by far not the most accurate diffusion model. For hydrogen flames, for instance, multicomponent diffusion models are much more accurate. For more details on this topic we refer to Braack & Ern [6] and the contribution [8] in this book where diffusion models are locally adapted.

In the numerical examples shown in this article we have used the simple diffusion law (7).

The temperature flux is modeled by Fourier’s law
\[ \mathcal{Q} = -\text{div} \lambda \nabla T, \]
M. Braack, and T. Richter

with the thermal conductivity $\lambda$. In this model the Dufour effect is not included, which is generally accepted to be of minor importance for the present flame configurations; see for instance Ern & Giovangigli [12]. The chemical production terms $\dot{\omega}_k$ are of the form

$$\dot{\omega}_k = \sum_{r \in R} (\nu'_{rk} - \nu_{rk}) k_r \prod_{s \in S} c_j^{\nu_{sj}},$$

with stoichiometric coefficients $\nu'_{rk}, \nu_{rk} \in \mathbb{N}$, and species concentrations $c_j = \rho x_j / m$. The temperature dependent reaction rate is of Arrhenius type

$$k_r = A_r T^{\beta_r} \exp \left\{ -\frac{E_r}{RT} \right\}.$$

The empirical coefficients $A_r, \beta_r$, and $E_r$ are given by the chemical mechanism. Furthermore, the coefficients $\lambda$ and $\mu$ are modeled by the average of the arithmetic and harmonic mean of their specific quantities,

$$\lambda := \frac{1}{2} \sum_{k=1}^{n_s} x_k \lambda_k + \frac{1}{2} \left( \sum_{k=1}^{n_s} \frac{x_k}{\lambda_k} \right)^{-1}, \quad \mu := \frac{1}{2} \sum_{k=1}^{n_s} x_k \mu_k + \frac{1}{2} \left( \sum_{k=1}^{n_s} \frac{x_k}{\mu_k} \right)^{-1},$$

with specific viscosities $\mu_k$ and heat conductivities $\lambda_k$, respectively. The heat capacity $c_p$ is given by

$$c_p := \sum_{k=1}^{n_s} w_k c_{pk}.$$

The coefficients $\mu_k, \lambda_k, D_{kl}^{bin}$ are given in terms of exponential fits of experimental data,

$$\mu_k := \sum_{j=0}^4 \alpha_{kj}^\mu T^j, \quad \lambda_k := \sum_{j=0}^4 \alpha_{kj}^\lambda T^j, \quad D_{kl}^{bin} := \sum_{j=0}^4 \alpha_{klj}^D T^j,$$

with coefficients $\alpha_{kj}^\mu, \alpha_{kj}^\lambda, \alpha_{klj}^D \in \mathbb{R}$. The specific enthalpies $h_k$ is given by similar exponential fit with 7 coefficients differently for the low and high temperature regime, $T \leq 1000 \, \text{K}$ or $T > 1000 \, \text{K}$, respectively. The specific heat capacity $c_{pk}$ is obtained by taking the derivative of $h_k$ with respect to the temperature:

$$c_{pk} = \frac{\partial h_{pk}}{\partial T} = \sum_{j=1}^{6} j \alpha_{kj}^h T^{j-1}.$$

Equations (1)–(4) are linearly dependent because the species mass fractions sum up to unity and

$$\sum_{k=1}^{n_s} F_k = \sum_{k=1}^{n_s} m_k \dot{\omega}_k = 0.$$
Therefore, we omit one equation in (4), 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

\[
p(x, t) = p_{th}(t) + p_{hyd}(x, t).
\]

While the so called thermodynamic pressure \( p_{th}(t) \) is constant in space, the hydrodynamic pressure part \( p_{hyd}(x, t) \) may vary in space and time. Hence, the pressure gradient in the momentum equation (2) can be replaced by \( \nabla p_{hyd} \). This is important for flows at low Mach number, because \( p_{hyd} \) is several magnitudes smaller than \( p_{th} \).

2 DISCRETIZATION BY STABILIZED FINITE ELEMENTS

In this section, we describe the weak formulation and the discretization of the governing equations. This provides the framework to derive the a posteriori error estimator used in an adaptive procedure.

2.1 Variational formulation

The vector \( u \) assembles the variables

\[
u := \{p_{hyd}, v, T, y_1, \ldots, y_s\}
\]

while the density is considered as a coefficient determined by the ideal gas law

\[
\rho = \frac{(p_{th} + p_{hyd})m}{RT}.
\]

By \( \hat{u} \) we denote an extension of non-homogeneous Dirichlet conditions to \( \Omega \). Hence, the solution is sought in the space \( \hat{u} + X \), where \( X \) is a function Hilbert space. The space \( X \) can be considered as a product of Hilbert spaces for each component, \( L^2(\Omega) \times H^1(\Omega)^{d+1+s} \), with standard modifications to build in the boundary conditions and probably restrictions on the mean of pressure. With the notation \((u, \varphi)\) for the usual \( L^2 \)-scalar product in \( \Omega \) we define the semilinear form for stationary solutions of equations (1)–(6):

\[
a(u)(\varphi) := (\text{div}(\rho v), \xi) + (\rho(v \cdot \nabla) v, \phi) + (\pi, \nabla \phi) - (p_{hyd}, \text{div} \phi) - (g, \phi) \\
+ ((\rho c_p v + \alpha) \cdot \nabla T, \sigma) + (\lambda \nabla T, \nabla \sigma) + \sum_{k=1}^{n_s} (h_k m_k \dot{\omega}_k, \sigma) \\
+ \sum_{k=1}^{s} (\rho v \cdot \nabla y_k - m_k \dot{\omega}_k, \tau_k) - (F_k, \nabla \tau_k)
\]
for test functions $\varphi = \{\xi, \phi, \sigma, \tau_1, \ldots, \tau_s\}$. With this notations $u$ is solution of the equation

$$
u \in \hat{u} + X : \quad a(u)(\varphi) = 0 \quad \forall \varphi \in X.$$  

### 2.2 Galerkin formulation

For the discretization we use a conforming equal-order Galerkin finite element method defined on quadrilateral (hexaedrons in 3D) meshes $T_h = \{K\}$ over $\Omega$, 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 $K$. In order to ease the mesh refinement we allow the cells 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 $V_h$ consists of continuous, piecewise polynomial functions (so-called $Q_1$-elements) for all unknowns,

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

where $Q_1(K)$ is the space of functions obtained by transformations of (isoparametric) bilinear polynomials from a fixed reference unit cell $\hat{K}$ to $K$. For a detailed description of this standard construction, see [11] or Johnson [15].

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 implementation details, see e.g. Carey & Oden [10].

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

$$u_h \in \hat{u} + X_h : \quad a(u_h)(\varphi) = 0 \quad \forall \varphi \in X_h. \quad (10)$$

The formulation (10) 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). This problem will be addressed in more detail in the following.

### 2.3 Drawbacks of standard stabilization techniques

The classical streamline diffusion (SUPG) stabilization for the incompressible Navier-Stokes problem, introduced by Hughes & Hughes [9], stabilizes the convective terms. Johnson & Saranen created in [16] an additional pressure stabilizing (PSPG) term in order to allow equal-order finite element approximations of velocity and pressure. Despite the success of the classical SUPG/PSPG stabilization technique for incompressible flow, one can find in recent papers a critical evaluation of this approach, see e.g. [22]. Drawbacks
are basically due to the strong coupling between velocity and pressure in the stabilizing terms and the nontrivial construction of efficient algebraic solvers. The treatment of time-dependent problems requires the usage of space-time elements. Furthermore for reactive flows with a large number of species, the introduction of strong couplings between different chemical species becomes a severe problem. For illustration, we consider a single stationary convection-diffusion-reaction for species $y_k$ equation of the form

$$
\beta \cdot \nabla y_k - \text{div} (D_k \nabla y_k) = f_k, \quad (11)
$$

with a source term $f_k = f_k(T, y_1, \ldots, y_n)$ depending on temperature and other chemical species. For simplicity we assume homogeneous Dirichlet conditions (also assumed to be included in $V_h$). The weak formulation stabilized with SUPG reads:

$$
y_k \in V_h : \quad (\beta \cdot \nabla y_k - f_k, \phi) + (D_k \nabla y_k, \nabla \phi) + \sum_{K \in T_h} (\beta \cdot \nabla y_k - \text{div} (D_k \nabla y_k) - f_k, \delta_K \beta \cdot \nabla \phi)_K = 0 \quad \forall \phi \in V_h,
$$

with cell-dependent parameters $\delta_K$. In the corresponding stiffness matrix, the mass fractions of different chemical species are coupled due to the zero-order term $(f_k, \phi)$, because $f_k = f_k(T, y_1, \ldots, y_n)$, and due to the stabilization term $(f_k, \delta_K \beta \cdot \nabla \phi)$. The first coupling includes only the degrees of freedom corresponding to the same mesh points when the mass matrix is lumped. The latter coupling includes also degrees of freedom from different mesh points and different chemical species. Hence, the SUPG stabilization enlarges the number of couplings substantially.

Therefore, we present an alternative stabilization technique firstly introduced by Becker & Braack [1] which avoids this kind of additional couplings.

### 2.4 Stabilization by local projection

The nonlinear problem stabilized by local projection is of the form:

$$
u_h \in \hat{u} + X_h : \quad a(u_h)(\varphi) + s_h(u_h)(\varphi) = 0 \quad \forall \varphi \in X_h. \quad (12)
$$

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

For the definition of $s_h(\cdot)(\cdot)$ we suppose that the triangulation $T_h$ is constructed in such a way that it results from a coarser quasi-regular mesh $T_{2h}$ by one global refinement. By a “patch” of elements we denote a group of cells (four cells in 2D) which results from a common coarser cell in $T_{2h}$. 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

$$
\kappa_h : V_h \to V_h, \quad \kappa_h \varphi := \varphi - I_{2h}^h \varphi
$$


we denote the difference between the identity and this interpolation. With this notation, the stabilization term added to the Galerkin formulation for an equation of type (11) reads

\[(\beta \cdot \nabla \kappa_h y_k, \delta_h \beta \cdot \nabla \kappa_h \phi),\]

where the parameter \(\delta_h\) is chosen cell-wise constant depending on the local balance of convection and diffusion:

\[\delta_h|_K := \frac{\delta_0 h^2_k}{6D_k + h_K \|\beta\|_{\infty,K}}.\]

Here, the quantity \(\|\beta\|_{\infty,K}\) is the maximum of \(\beta\) on cell \(K\). The parameter \(\delta_0\) is a fixed constant, usually chosen as \(\delta_0 = 0.5\). Note, that \(\kappa_h\) vanishes on \(V_{2h}\), and therefore, the stabilization vanishes for test functions of the coarse grid \(\phi \in V_{2h}\).

The velocity-pressure coupling is stabilized for equal-order elements by a diffusive term acting on the pressure fluctuations:

\[\left(\nabla \kappa_h p_h, \alpha \nabla \kappa_h \xi\right).\]

For a stability proof and an error analysis for the Stokes equation we refer to [1].

For the full reactive flow system this stabilization technique is applied to all convective terms:

\[s_h(u_h)(\varphi) := \left(\nabla \kappa_h p_h, \alpha_h \nabla \kappa_h \xi\right) + \left((\rho_h v_h \cdot \nabla) \kappa_h v_h, \delta_v, h(\rho_h v_h \cdot \nabla) \kappa_h \phi\right) + \left((\rho_h v_h \cdot \nabla) \kappa_h T_h, \delta_T, h(\rho_h v_h \cdot \nabla) \kappa_h \sigma\right) + \sum_{k=1}^s \left((\rho_h v \cdot \nabla \kappa_h y_k, \delta_k \rho_h v_h \cdot \nabla \kappa_h \tau_k\right),\]

The proposed stabilization is consistent in the sense that the introduced terms vanish for \(h \to 0\) with the correct order. Let us briefly 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.

3 LOCAL MESH REFINEMENT

Adaptive mesh refinement is well known to be an efficient method to increase the accuracy of the discrete solution with respect to the number of degrees of freedom, especially for an accurate resolution of a flame front. The efficiency of local mesh refinement strongly depends on the refinement strategy and the information about discretization errors. The *a posteriori error* estimates based on weighted residuals, introduced by Becker and Rannacher [3, 4] allows for a posteriori error control with respect to functional output \(j(u)\).
For a detailed description we refer to the contribution [8] and the references cited therein. Here, we simply state the estimator used in the simulations presented below:

\[ |j(u) - j(u_h)| \approx \sum_{i=1}^{n} \eta_i, \quad \eta_i := |a(u_h)(\varphi_i^{(2)})Z_i^\pi|, \]

where \( \varphi_i^{(2)} \) denotes the \( Q_2 \) basis function corresponding to the node \( i \) of the mesh, and \( Z_i^\pi \) is a coefficient obtained by solving the linear dual problem

\[ z_h \in X_h : \quad a_h^*(u)(\varphi, z_h) = j'(u_h)(\varphi) \quad \forall \varphi \in X_h. \]

with the bilinear form

\[ a_h^*(u)(\varphi, z_h) := \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} \left\{ a(u_h + \varepsilon \varphi, z_h) + s_h(u_h + \varepsilon \varphi, z_h) - a(u_h)(z_h) - s_h(u_h)(z_h) \right\} \]

Since this equation is linear, the computational cost of its solution relates to the cost of one Newton step in solving the primal problem. We note that the solution of the dual problem on the same mesh and by the same method as used for the primal problem is not mandatory. One may use a coarser mesh to save computational cost or even employ another (possibly higher-order) discretization to get the weights with enhanced accuracy. In our computations, the dual solution is always computed by the same method as used for the primal problem.

4 SOLUTION PROCEDURE

The discrete equation system (12) is 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 LU factorization can be applied. This accounts for the strong local coupling of hydrodynamical variables as pressure, velocity and temperature with the chemical variables. This linear solver is described in detail in the contribution [2] of this book. 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 the form

\[
\begin{bmatrix}
A_{pp} & A_{pv} & A_{pT} & A_{py} \\
A_{vp} & A_{vv} & A_{vT} & A_{vy} \\
A_{Tp} & A_{Tv} & A_{TT} & A_{Ty} \\
A_{yp} & A_{yv} & A_{yT} & A_{yy}
\end{bmatrix}
\]
Table 1: Memory needed for storing one system matrix in single precision and $s = 20$ species in two and three dimensions.

<table>
<thead>
<tr>
<th></th>
<th>2D</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>matrix couplings</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td>block size</td>
<td>576</td>
<td>625</td>
</tr>
<tr>
<td>cells necessary for $\approx 5%$ error</td>
<td>10,000</td>
<td>250,000</td>
</tr>
<tr>
<td>memory (single prec.)</td>
<td>52 MB</td>
<td>4.2 GB</td>
</tr>
</tbody>
</table>

Note, that the computational effort is proportional to the number of matrix entries. As already mentioned in Section 2, 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 1 we show the memory necessary for storing one system matrix in 2D and 3D when a reaction mechanism with $s = 20$ 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 couplings do not appear in off-diagonal blocks $J_{ij}$, $i \neq j$.

- Some diffusion laws, as for instance the extended Fick’s law (8), show off-diagonal couplings when mass fractions $y_k$ are used as primary variables. Even more complex diffusion models, e.g., multicomponent diffusion, generate off-diagonal couplings even when mole fractions $x_k$ are used. 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 (7) this is the case without modification of the Jacobian.

- SUPG stabilization generate off-diagonal couplings between species gradients due to the consistency 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 form (14) while the off-diagonal blocks $J_{ij}$, $i \neq j$ of the system matrix are stored in the form

$$J_{ij} = \begin{bmatrix} A_{pp} & A_{pv} & A_{pT} \\ A_{vp} & A_{vv} & A_{vT} \\ A_{Tp} & A_{Tv} & A_{TT} \\ \end{bmatrix} D_{yy}$$
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 $s = 20$ species, the memory needed to store a matrix is a factor

$$[27(5 + s)^2] : [(5 + s)^2 + 26(5^2 + s)] \approx 13.5$$

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 details we refer to [5] and [7].

5 SIMULATION OF A 2D BURNER

The aim is the numerical simulation and its comparison with experimental data of a partially premixed methane flame at atmospheric pressure carried out at the Institute of Physical Chemistry (PCI) in Heidelberg. In particular, it was possible to measure absolute concentrations of formaldehyde ($CH_2O$) for the stationary laminar flame. A numerical simulation of such minor species poses high demands on the spatial accuracy of the discretization and on the reaction mechanism, as well.

Since the burner consists of seven identical slots it has a certain two dimensional structure, see the photo in Figure 1. Its capacity and energy density is comparable to a normal household burner. A numerical simulation of such a burner is a good test for the performance of the numerical method. Under the assumption of symmetry we restrict the simulation to one slot of the burner. Two simulations with different chemical mechanisms are performed:

- a so-called C1 mechanism with 15 chemical species and 42 reversible reactions of Smooke, see [21], and

- a C2 mechanism with 39 chemical species and 151 reversible reactions of Warnatz [24]. This mechanism includes formation of several chemical species containing two carbon atoms.

The adaptation process is based on the mean value of formaldehyde mass fractions along the vertical line $\Gamma$ at the center of the slot,

$$j(u) := \int_{\Gamma} y_{CH_2O},$$

and produces a mesh with about $10^4$ nodes. The corresponding CPU time on a single PC with 1 GHz was 4 hours for the small mechanism. In Figure 2 we show several species
obtained by this simulation. A tensor grid with $10^6$ nodes would give a similar accuracy, but about 1 week of CPU time would have been necessary.

Hence, the economy of the discretization gives the possibility to use the more exact C2 mechanism. The already obtained solution serves as a good initial guess for the Newton iteration. However, the mesh is adapted once more, because the flame front moves upstream. This can also be seen in Figure 3 where a cross-section of formaldehyde is compared with the experiments. The C2 mechanism is obviously more accurate.

After 23 hours CPU time we get the stationary solution with 39 chemical species. Figure 4 shows a zoom of the obtained result for ethylidyne $C_2H_3$, a species which is not included in the smaller system. On the left hand side of the figure, the locally refined mesh is shown.
6 SIMULATION OF A 3D BURNER

Even more challenging is the computation of three-dimensional flames. The household burner constructed by BOSCH is an example of a burning facility with a three-dimensional laminar stationary flame, see Figure 5. Again this burner consists of several slots, but in contrast to the previous configuration, the symmetry is violated due to several cooling ducts transversal to the lamella. Within the PhD project [18], a two-dimensional approximation was arranged by neglecting the cooling ducts. With the software GASCOIGNE [13] a parameter study was carried out in order to obtain information concerning pollution formation under several loading situations of the burner, see [19].

As a first step towards three-dimensional combustion simulations we use the C1 and C2 reaction mechanism of the previous section with 15 and 39 chemical species, respectively. Due to the large number of chemical species, the nonlinearities 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.

6.1 Two dimensional starting guess with C1 mechanism

The three dimensional simulation is started with a prolongation of the solution of a two dimensional (2D) simplification. In Figure 7, cut-outs of adaptively refined meshes
from this 2D simplification are given. The mesh adaptation is driven by the adaptive process described before.

As quantity of interest we aim at detecting the concentration of formaldehyde $CH_2O$ along a horizontal line through the three dimensional domain. In the two dimensional simplifications this quantity is represented by the evaluation at 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.

### 6.2 Three dimensional computation with C1 mechanism

In Figure 6 the computational domain including the cooling duct is described. The three dimensional simulations are performed on a PC cluster with the techniques described in this work and the parallelization technique presented in detail in the contribution [2] in this book. 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 and will be analyzed later in this section.

In order to obtain a stationary solution the flame front has to be resolved sufficiently. Since the size of the flame front is determined by the species diffusion, an upscaling of the diffusion fluxes by a homotopy parameter $\tau \geq 1$ smears the flame front. For $\tau = 3$, a stationary solution is obtained even on the coarsest mesh with 26008 cells only. On finer meshes the homotopy parameter is decreased down to 1 in order to recover the original diffusion model. The development of $\tau$ is listed in Table 2.

In Table 3, the performance of the parallel solver is summarized. The finest mesh used
corresponds to more than 11 million degrees of freedom. On 105 CPU’s (AMD Athlon, 1.4 GHz) the solver needs about 22 minutes for one time step on this mesh. On the coarsest mesh, one time step lasts about 16 minutes on 32 processors. Since several time steps were necessary, 6.5 hours are needed on this mesh in order to get the Newton iteration converged.

The computation is performed on a large Linux cluster with several tasks of different users at the same time. On those multi-user machines, it is in general not possible to have always the optimal number of CPU for this computation. Therefore, the number of CPU’s is not necessarily increasing for larger problem sizes.

The problem is by far too large to be computed on a single processor, and hence the parallel efficiency can not be determined. However, an appropriate measure for the parallel efficiency is

\[
\text{par eff} := 100 \frac{\#\text{CPU} \cdot \text{time}}{\text{dof}}
\]

This quantity is listed in the last column. This efficiency becomes better for larger problem size.

Table 2: Evolution of the homotopy parameter \(\tau\) in dependence of the number of cells. For \(\tau > 1\) the flame front is broader so that even on coarser meshes a stationary solver can be applied. For \(\tau = 1\) the Fick’s law is recovered. This 3D computation uses the C1 mechanism.

<table>
<thead>
<tr>
<th>cells</th>
<th>dof’s</th>
<th>(\tau)</th>
</tr>
</thead>
<tbody>
<tr>
<td>26008</td>
<td>574408</td>
<td>3.0</td>
</tr>
<tr>
<td>65880</td>
<td>1415842</td>
<td>2.8</td>
</tr>
<tr>
<td>185020</td>
<td>3785509</td>
<td>2.5</td>
</tr>
<tr>
<td>558694</td>
<td>11353332</td>
<td>1.0</td>
</tr>
</tbody>
</table>
6.3 Comparison of the 2D and 3D solutions

In Figure 8, the three-dimensional effect due to the cooling tubes are clearly visible. In particular, the flow velocity is reduced above the cooling tubes and the flame front becomes less pronounced. The visualization is done with the software VisuSimple [23].

For analyzing the three dimensional character of the burner – particularly the influence of the cooling ducts – we look at the course of the flame front along the z-axis of the burner. In Figure 9, we plot the process of the already mentioned functional evaluated along a line through the burner. This line is located in the flame front and crosses the domain atop of the cooling devices. In Figure 9, we show the velocity in main flow direction, the mass fraction of formaldehyde $CH_2O$ and of $HO_2$ radicals. In addition, we plot the value of the corresponding solution component in the specific point obtained with the two dimensional simplification.

The plotted quantities feature a significant variation in z-direction, but also an overall discrepancy in comparison to the 2D simplification is observed. In Table 4 we list the maximal values of these four components identified in the whole domain for the two and the three dimensional setting. The difference in the temperature of about 100 Kelvin is quite remarkable and is caused by the cooling device. The lower temperature in the 3D-configuration leads to large differences in all other components. Taking in mind that the production of the pollutant $NO_x$ is extremely sensitive to temperature it is questionable that a two-dimensional simulation is appropriate for an accurate prediction of $NO_x$.

Summarizing the results, the considered configuration yields real three-dimensional features. Two-dimensional simplifications of the geometry are not valid. However, detailed
three-dimensional simulations are possible, if one combines efficient solvers with mesh adaptation and parallelization.

We refer to [20] for more details on the configuration and the comparison between the two- and three-dimensional simulations.

6.4 3D simulation with 39 chemical species

The final computation breaks the dimensions of all the previous computations. We perform a three dimensional simulation of the burner with the C2 mechanism (39 chemical species). The solution looks quite similar to the C1 solution but looking on numbers, there is indeed a difference. In Table 5 some maximum values are listed in order to compare the effect of the dimension (2D or 3D) and the reaction mechanism used. While the mechanism has nearly no effect on the maximum temperature, the impact on \( CH_2O \) and \( HO_2 \) is quite large.

The homotopy parameter is taken as \( \tau = 3 \) on the coarsest mesh and reduces to \( \tau = 1 \) on the finest mesh. Hence, we recover the original Fick’s law. The performance of the parallel solver is summarized in Table 6.

On the finest mesh we have more than 14 millions degrees of freedom. The solver needs about 78 minutes for one time step when 18 processors (Opteron CPU’s) are used. Due to the sparse matrix structures designed for these type of problems only 25 GB of total memory is necessary.

Table 5: 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 for both reaction mechanisms.

<table>
<thead>
<tr>
<th></th>
<th>C1 mechanism</th>
<th>C2 mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>velocity (m/s)</td>
<td>3.39</td>
<td>2.61</td>
</tr>
<tr>
<td>temperature (K)</td>
<td>2156</td>
<td>2039</td>
</tr>
<tr>
<td>mass fraction ( CH_2O \times 10^{-3} )</td>
<td>7.1</td>
<td>5.2</td>
</tr>
<tr>
<td>mass fraction ( HO_2 \times 10^{-4} )</td>
<td>2.3</td>
<td>6.2</td>
</tr>
</tbody>
</table>

Table 6: Calculations for the C2 mechanism done on an Opteron Linux clusters. On different meshes the number of cells, the number of degrees of freedom, the time necessary to solve one time step, the number of CPU’s, the memory usage and the parallel efficiency index are given. Further the used homotopy parameter \( \tau \) is indicated.

<table>
<thead>
<tr>
<th>cells</th>
<th>dof’s</th>
<th>#CPU</th>
<th>memory (GB)</th>
<th>time (sec)</th>
<th>( I_{eff}^{par} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 872</td>
<td>825 084</td>
<td>3</td>
<td>1.10</td>
<td>1772</td>
<td>0.64</td>
</tr>
<tr>
<td>29 648</td>
<td>1 505 946</td>
<td>5</td>
<td>2.23</td>
<td>1583</td>
<td>0.53</td>
</tr>
<tr>
<td>60 784</td>
<td>3 136 248</td>
<td>10</td>
<td>5.26</td>
<td>2248</td>
<td>0.71</td>
</tr>
<tr>
<td>197 816</td>
<td>9 863 082</td>
<td>10</td>
<td>16.92</td>
<td>6096</td>
<td>0.61</td>
</tr>
<tr>
<td>291 102</td>
<td>14 299 478</td>
<td>18</td>
<td>25.25</td>
<td>4681</td>
<td>0.59</td>
</tr>
</tbody>
</table>
Figure 8: Simulation of the three-dimensional methane burner: (a) Vertical velocity, (b) temperature, (c) $H$ mass fractions, and (d) $OH$ mass fractions.
Figure 9: 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.

REFERENCES


