High-Quality Real-Time Simulation of a Turbulent Flame

Piotr Opioła

Institute of Computer Science, Jagiellonian University
Łojasiewicza 6, 30-348 Kraków, Poland

Abstract. We consider a 2-dimensional model of a turbulent, methane flame. The model is based on the compressible Navier-Stokes equations extended with a temperature equation and chemical reaction properties. In order to achieve a high-quality real-time simulation, a fully adaptive finite element method is used to solve the considered system of equations. The method performs adaptive mesh refinement, local adjustment of the approximation order, and multilevel adaptation. The structure and composition of the flame, along with the numerical properties of the method, such as the mesh density, are studied. The results are compared to results obtained with a direct numerical simulation.

Keywords: combustion, flame, Navier-Stokes equations, adaptive finite element method.

1 Introduction

Flame behavior has been extensively studied e.g. for the purpose of jet engine design [3]. Flame and smoke propagation models are used in fire-protection systems [4]. Simplified flame models are also of interest of the entertainment industry [8], [10]. In this paper, we present a flame simulation technique that can be used in either case. We use a 5-step reduced mechanism of methane-oxygen combustion, based on [13], which gives already reliable results while remaining relatively simple. The dynamics of the system is modeled with a variation of Navier-Stokes equations, presented e.g. in [5].

The model equations are solved with an adaptive finite element method. It is common to use adaptive mesh refinement for flame simulation, e.g. [5], [2], [1]. The refinement is usually performed to obtain high-resolution flame fronts. We propose an algorithm which additionally performs coarsening of the mesh at the regions of fast flow in order to obtain stability of the algorithm without decreasing the time step length. Further improvement of the convergence rate of the method is achieved by adjustment of the approximation order [12]. In order to keep track of small-scale features of the flow, we use a multilevel adaptation [15]. All the adaptive techniques are used to control the approximation error and ensure the stability of the algorithm.

B. Murgante et al. (Eds.): ICCSA 2011, Part IV, LNCS 6785, pp. 112–122, 2011.

2 Model

The main model equations are [5]

$$\frac{\partial}{\partial t}\rho_i + \operatorname{div}(\rho_i v) = \operatorname{div}(\rho \mathsf{D}_i \nabla Y_i) + \omega_i,\ i = 1 \dots N_s\,, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho v) + \operatorname{div}(\rho v \otimes v) = -\nabla p + f_b\,, \tag{2}$$

$$\frac{\partial}{\partial t}(\rho T) + \operatorname{div}(\rho T v) = \frac{1}{c_p}\left(\operatorname{div}(\lambda \nabla T) - \sum_{i=1}^{N_s} \omega_i h_i\right)\,, \tag{3}$$

where ρ_i is the density of the i-th species, ρ the density of the mixture, v the velocity vector, p the pressure, T the temperature, Y_i the mass fraction of the i-th species, ω_i the reaction rate of the i-th species, N_s the number of species, f_b the buoyancy force, h_i the specific enthalpy of the i-th species, D_i the diffusion coefficient of the i-th species, c_p the specific heat at constant pressure, and λ the conduction coefficient. All the variables are functions defined on $\Omega \times [0, t_{max}]$, where $t_{max} < \infty$, and $\Omega \subset \mathbb{R}^2$. The mass fraction is given by

$$Y_i = \frac{\rho_i}{\rho}\,. \tag{4}$$

Pressure is given by the ideal gas law

$$p = R\rho T\,, \tag{5}$$

where R is the individual gas constant of the mixture. The specific enthalpy is given by

$$h_i = h_i^0 + \int_{T_0}^{T} c_{p,i}\, dT\,, \tag{6}$$

where h_i^0 is the standard enthalpy of formation of the i-th species, T_0 the standard temperature, and $c_{p,i}$ the specific heat of the i-th species at constant pressure.

The reaction properties are given by [7]

$$\omega_i = \sum_{j=1}^{N_r} \nu_{ij} q_j\,, \tag{7}$$

$$q_j = A_j T^{\beta_j} e^{\frac{-E_{aj}}{R_0 T}} \prod_{i=1}^{N_s} \left(\frac{\rho_i}{M_i}\right)^{-min\{\nu_{ij},0\}}\,, \tag{8}$$

where N_r is the number of reactions, q_j the rate of progress for the j-th reaction, $A_j > 0$ the pre-exponential factor for the j-th reaction, β_j the temperature exponent for the j-th reaction, ν_{ij} the stoichiometric coefficient of the i-th species for the j-th reaction, $E_{aj} > 0$ the activation energy for the j-th reaction, R_0 the

gas constant, and $M_i > 0$ the molar mass of the i-th species. The stoichiometric coefficients ν_{ij} are positive for products, and negative for reactants, so the following equation holds

$$\sum_{i=1}^{N_s} \omega_i = 0\,. \tag{9}$$

The chemical reactions are

$$\mathrm{CH_4} \rightarrow \mathrm{CH_3} + \frac{1}{2}\mathrm{H_2}\,, \tag{10}$$

$$2\mathrm{CH_3} + \frac{1}{2}\mathrm{O_2} \rightarrow \mathrm{CO} + 3\mathrm{H_2} + \mathrm{C}\,, \tag{11}$$

$$\mathrm{C} + \frac{1}{2}\mathrm{O_2} \rightarrow \mathrm{CO}\,, \tag{12}$$

$$\mathrm{CO} + \frac{1}{2}\mathrm{O_2} \rightarrow \mathrm{CO_2}\,, \tag{13}$$

$$\mathrm{H_2} + \frac{1}{2}\mathrm{O_2} \rightarrow \mathrm{H_2O}\,. \tag{14}$$

The mechanism is based on the 7-step mechanisms listed in [13] for lean, stoichiometric and rich methane-air mixtures. The first three reactions, in the case of a stoichiometric mixture, could be replaced by $\mathrm{CH_4} + \frac{1}{2}\mathrm{O_2} \rightarrow \mathrm{CO} + 2\mathrm{H_2}$. Splitting this into three reactions is important especially in the case of a lean mixture, when methyl takes part in soot formation in the 2nd reaction. In the 3rd reaction, by further oxidation, the soot forms carbon monoxide, which turns into carbon dioxide in the 4th reaction. Finally, the remaining hydrogen reacts with oxygen to produce water. Noticeably, all the reactions except the first consume oxygen, and thus limited oxygen supply may lead to release of soot and carbon monoxide as products of an incomplete combustion. The values of the reaction rate coefficients are given in Table 1.

Table 1. The reaction rate coefficients, calculated from GRI-Mech [14]

Reaction number	A_j	β_j	E_{aj} (cal/mol)
1	6.6E+8	1.6	10840
2	4E+13	0	0
3	5.8E+13	0	576
4	1.8E+10	0	2385
5	3.87E+4	2.7	6260

The initial and boundary conditions are as follows. A methane-oxygen mixture is injected into the model domain from the bottom, as in Figure 1. For all the boundary, we use Dirichlet boundary conditions: $p = const$, $\rho_i = const$. The initial temperature at the place of injection is raised above the minimum autoignition temperature.

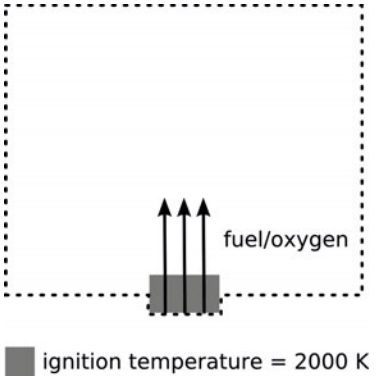

Fig. 1. The initial and boundary conditions

3 Algorithm

The model equations are solved with a fully adaptive finite element method, based on [11]. We define a mesh as a finite set $\mathcal{M} = \{x_i\}_{i \in I \subset \mathbb{N}} \subset \Omega$ and a neighborhood function $\mathcal{N} : \Omega \to \mathcal{P}(\mathcal{M})$. The neighborhood function maps a point $x \in \Omega$ into a nonempty set of its neighbors. The set of neighbors is chosen to minimize the condition

$$\gamma \sum_{x_i \in \mathcal{N}(x)} \|x - x_i\| + \max_{x_i \in \mathcal{N}(x)} \min_{x_j \in \mathcal{N}(x)} \measuredangle(x_i, x, x_j)\,, \tag{15}$$

where $\gamma > 0$ is a constant, $\|\cdot\| \equiv \|\cdot\|_2$ is the Euclidean norm, and $\measuredangle(x, y, z)$ is equal to the angle $\alpha \in [0, 2\pi]$ between $\overline{xy}$ and $\overline{yz}$. If $x = y$ or $y = z$ the angle is assumed 0. If $x = z \neq y$ the angle is assumed 2π.

The domain is divided into elements. Two points x, y belong to the same element iff $\mathcal{N}(x) = \mathcal{N}(y)$. This means that every mesh point itself is a degenerate element.

Having a mesh $(\mathcal{M}, \mathcal{N})$, we define the approximation $\tilde{u} : \Omega \to \mathbb{R}$ as

$$w_i(x) = 1 - (|\mathcal{N}(x)| - 1)\frac{\|x - x_i\|}{\sum_{x_j \in \mathcal{N}(x)} \|x - x_j\|}\,, \tag{16}$$

$$\Phi_i(x) = \frac{w_i(x)^{p_i}}{\sum_{x_j \in \mathcal{N}(x)} w_j(x)^{p_j}}\,, \tag{17}$$

$$\tilde{u}(x) = \sum_{x_i \in \mathcal{N}(x)} \Phi_i(x)\,(u_i + f_i \cdot (x - x_i))\,, \tag{18}$$

where $p_i \in \mathbb{R}$, $u_i \in \mathbb{R}$, $f_i \in \mathbb{R}^2$ are the approximation parameters. The parameters, as well as the mesh points $\mathcal{M}$, are sought to minimize the monitor function E, defined as

$$E = \frac{1}{a} E_\varepsilon + \frac{1}{b} E_I + E_\delta\,, \tag{19}$$

where

$$E_\varepsilon = \max_{x\in\Omega} \|\tilde{u}(x) - u(x)\|\,, \tag{20}$$

$$E_I = |I| = |\mathcal{M}|\,, \tag{21}$$

$$E_\delta = \max_{i\neq j} \frac{1}{\|x_i - x_j\|}\,, \tag{22}$$

$u : \Omega \to \mathbb{R}^2$ is the approximated function, and a, b are parameters of the method. An approximation analogous to the above can be defined for a vector function as well, that is for $u : \Omega \to \mathbb{R}^2$. The parameters a and b have the meanings of the expected error and the expected number of mesh points respectively. The parameters provide a convenient way to control the balance between the precision and the efficiency of the algorithm.

In the considered flame model, there are $N + 2$ variables: ρ_i, v, and T. For each variable, we seek for a solution, i.e. a mesh and an approximation defined on the mesh. The algorithm goes as follows

1. Generate a regular mesh on Ω for each model variable, $t \leftarrow 0$
2. Set the initial values
3. Refine each mesh by adding a new point in the center of every non-degenerate element and at each free node of every element
4. Solve the model equations on the given meshes
5. Coarsen the meshes by removing unimportant points
6. Add sub-levels to the meshes where the mesh density increases over an acceptable value
7. If the monitor function did not change significantly for any of the meshes: $t \leftarrow t + \Delta t$
8. Go to step 3

A free node of an element is a node which is not a mesh point. Deciding whether or not a mesh point is important is done by comparing the value of the monitor function before and after removing the point. If the value decreases, the point is unimportant. Otherwise, the point has to be restored. A significant change of the monitor function, mentioned in step 7 of the algorithm, is indicated by $|E^i - E^{i-1}| > \epsilon$, where i is the current iteration number and $\epsilon > 0$ is an arbitrary constant. The approximation error E_ε is estimated by the formula

$$E_\varepsilon \approx \max_{i\in I} \max_{x_j\in\mathcal{N}^\star(x_i)} \|\tilde{u}(\frac{x_i + x_j}{2}) - \tilde{u}^\star(\frac{x_i + x_j}{2})\|\,. \tag{23}$$

The star * denotes values calculated for $\mathcal{M}\backslash\{x_i\}$, that is without considering x_i as a mesh point.

3.1 Finding the Neighbors and the Elements

The set of neighbors $\mathcal{N}(x)$ for a given point $x \in \Omega$ and a given mesh $\mathcal{M}$ is found in constant average time. We use a Cartesian grid to localize quickly the

close mesh points in a given area. The neighbors are first sought in the grid cell which contains x. Next, the coincident cells are checked, and so on, until it is clear that no better set of neighbors can be found in respect to the optimization condition (15). For every checked grid cell, all the mesh points in the cell are tested as neighbors. If adding a given mesh point to the set of neighbors decreases (15), the point stays in the set. Otherwise, it is removed. If, for a current set of neighbors, the highest possible gain is lower than the lowest possible loss, the search ends. The side length of a grid cell is set to $\sqrt{\frac{|\Omega|}{|I|}}$. The parameter γ is set to $\frac{\pi}{6d} \ln |I|$, where d is the average distance between the mesh points. This way, the expected number of mesh points in a single grid cell is 1, and the expected number of neighbors for a given point is 3. Thus checking up to 9 grid cells is usually sufficient to find the set of neighbors. Generating the Cartesian grid and placing the mesh points in the correct cells takes linear time in respect to the size of the mesh, but it is only done once for a given mesh.

The elements are found in linear time in respect to the number of them. The edges of the elements are given by (x_i, x_j), $x_i \in \mathcal{M}$, $x_j \in \mathcal{N}^\star(x_i)$. Some of the edges may intersect. At each intersection a new element node is added, and the intersected edges are split. To find the intersections in linear time, again we use the Cartesian grid. Only the edges that share at least one coincident cell are tested for intersection. Once the element edges are establish, we can calculate the center points of the elements as the geometric centers of their nodes.

3.2 The Numerical Solver

Each model equation is a convection-diffusion equation, and thus it can be written in the general form

$$\frac{\partial}{\partial t}(\rho u) + \operatorname{div}(\rho u v) = \operatorname{div}(\mathsf{D}\nabla u) + s\,, \tag{24}$$

where u is a model variable in an intensive form (Y_i, v or T). This can be transformed to

$$\frac{\partial}{\partial t}(\rho u) + \rho v \cdot \nabla u + u v \cdot \nabla \rho + \rho u \operatorname{div} v = \mathsf{D}\Delta u + \nabla D \cdot \nabla u + s\,, \tag{25}$$

where s is the source term. After rearrangement the equation reads

$$(\rho v - \nabla \mathsf{D}) \cdot \nabla u = \mathsf{D}\Delta u + s - \frac{\partial}{\partial t}(\rho u) - u v \cdot \nabla \rho - \rho u \operatorname{div} v\,. \tag{26}$$

Let us introduce the following additional notation

$$f := \mathsf{D}\Delta u + s - \frac{\partial}{\partial t}(\rho u) - u v \cdot \nabla \rho - \rho u \operatorname{div} v\,, \tag{27}$$

$$g := \rho v - \nabla \mathsf{D}\,. \tag{28}$$

The equation (26) is solved in the following steps

1. $u_i^* \leftarrow u_i$, $i \in I$
2. $f_i^* \leftarrow \frac{f(x_i)}{|g(x_i)|} g(x_i)$, $i \in I$
3. $u_i^* \leftarrow \tilde{u}^\star(x_i)$, $i \in I$
4. Find p_i^* that minimizes $\| \frac{f(x)}{|g(x)|} g(x) - \nabla \tilde{u}^*(x) \|$, $i \in I$, $x := \frac{x_i + x_j}{2}$, $x_j \in \mathcal{N}^\star(x_i)$
5. If any of the approximation parameters changed significantly, go to step 2

The asterisk * denotes values calculated for the next time step. All the derivatives in f and g are approximated with the basic schemes. The time derivative in f is approximated using $\tilde{u}^*$ as

$$\frac{\partial}{\partial t}(\rho u) \approx \frac{\tilde{\rho}^* \tilde{u}^* - \tilde{\rho} \tilde{u}}{\Delta t} \,. \tag{29}$$

3.3 Multi-level Approach

If the following condition holds for any $x_i, x_j \in \mathcal{M}$, $x_i \neq x_j$

$$\| x_i - x_j \| < \delta \,, \tag{30}$$

where $\delta > 0$ is an arbitrary constant, then a new mesh level is added between x_i and x_j. The new level domain is defined as the convex hull of $\mathcal{N}(\frac{1}{2}(x_i + x_j))$. The calculation for every level is analogous to calculations for the root level. The height of the level tree is limited by an arbitrary constant.

3.4 Ensuring Real-Time Simulation

The simulation is said to run in real time if the time spent for calculating one time step is not greater than the time step length, $\Delta t_{\text{real}} \leq \Delta t$. This property is ensured by adjusting the method parameters a, b at every time step according to

$$b \leftarrow b \frac{\Delta t}{\Delta t_{\text{real}}} \,, \tag{31}$$

$$a \leftarrow \frac{a}{b} \,. \tag{32}$$

The initial values of the parameters are $a = \epsilon$, $b = 1$.

4 Results

We have tested the algorithm on three test cases corresponding to different equivalence ratios of the methane-oxygen mixture: $\phi = 0.6$ (a lean mixture), $\phi = 1$ (a stoichiometric mixture) and $\phi = 1.5$ (a rich mixture). The method accuracy has been verified by comparing the results to the those obtained with the direct numerical simulation and to the results presented in [2], [5], [6], [1] and [9]. The structure and composition of the flame, along with the numerical properties of the method, such as the mesh density, have been studied.

The mixture is injected into the domain with speed of 30 cm s^{-1}, through a hole with area of 12.56 mm^2. The temperature at the ignition area is set to 2000K at the beginning of the simulation. The numerical error tolerance ϵ is set to 10^{-10}, the minimum distance between mesh points δ is set to 10^{-3}, and the time step length Δt is set to 0.05s. The size of the domain is 20cm × 30cm.

Table 2. Frames per second (FPS), the number of mesh points, the error estimate, and the error with respect to direct numerical simulation for various configurations of the method parameters, at $t = 5$s.

ϕ, a, b	FPS	$\|I\|$	E_ε	DNS error
$\phi = 0.6$				
a = 1E-10, $b = 1$	15.2	3303	1E-6	0.3086
$a = 0.01$, $b = 1$	138.2	69	0.0773	0.6026
a = 1E-10, $b = 1000$	23.5	2410	0.0001	2.398
$a = 0.01$, $b = 1000$	45.6	735	0.0025	9.573
$\phi = 1$				
a = 1E-10, $b = 1$	19.8	2836	1E-7	0.2579
$a = 0.01$, $b = 1$	150.5	63	0.0238	0.5496
a = 1E-10, $b = 1000$	15.9	3508	1E-8	0.1162
$a = 0.01$, $b = 1000$	60.3	612	0.0002	2.4803
$\phi = 1.5$				
a = 1E-10, $b = 1$	20	2018	1E-7	0.2112
$a = 0.01$, $b = 1$	151.2	66	0.0639	0.597
a = 1E-10, $b = 1000$	15.4	2922	1E-7	0.0769
$a = 0.01$, $b = 1000$	63.1	585	0.0003	1.0587

Table 2 shows how some of the simulation properties depend on the method parameters for the three test cases. We can see that the simulation speed, expressed in frames per second, is highly correlated with the number of mesh points, as well as the approximation error estimate. In most cases, the simulation speed is above the critical 16 frames per second. The lowest approximation error estimate corresponds to the lowest error with respect to DNS, even though the correlation between these two properties is generally weak. Better accuracy is obtained for higher equivalence ratios.

Table 3 describes the mesh properties with respect to values of the equivalence ratio. Surprisingly, the average number of sub-levels is highest in the stoichiometric case and the lowest in the lean mixture case. This can be explained by higher concentration of error in the case of a lean flame. For the stoichiometric mixture, the maximum error is lower but there are more high-error points. The maximum depth of the level tree is limited to 4 and it reaches the limit. Most of the mesh points in the lean- and rich-mixture cases have 3 neighbors, with 5 being the maximum. In the stoichiometric case, most of the mesh points have 6

Table 3. The mesh properties

Property	$\phi = 0.6$	$\phi = 1$	$\phi = 1.5$
Avg. number of sub-levels	95	144	116
Max. level tree depth	4	4	4
Avg. number of neighbors	3.54	5.37	3.8
Max. number of neighbors	5	12	5
Avg. number of elements	6506	11244	3936

neighbors. The average number of elements shows a strong correlation with the number of points and the average number of neighbors.

Table 4. The peak values of soot and carbon monoxide mass fractions in the post-flame zone

ϕ	Our results		Values from [9]	
	Y_{CO}	Y_{C}	Y_{CO}	Y_{C}
0.6	0.2	0.001	0.05	0
1	0.01	0	0.002	0
1.5	0.3	0.002	0.1	0.002

Fig. 2. The CO_2 density for lean, stoichiometric and rich methane-oxygen mixtures

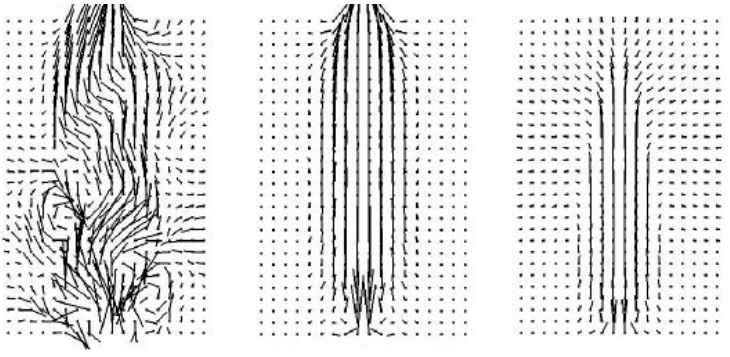

Fig. 3. The velocity vector field for lean, stoichiometric and rich methane-oxygen mixtures

Fig. 4. The light radiation in gray scale for lean, stoichiometric and rich methane-oxygen mixtures. The black color corresponds to the strongest radiation.

Table 4 compares the soot and CO release results with values presented in [9]. We can see that the soot release is highest in the rich-mixture case. Similar behavior is observed for carbon monoxide. Rich-mixture case is related to lower oxygen supply, which results in lower reaction rates of the reactions (2), (3), (4) and (5).

The CO_2 density presented on Figure 2 indicates the location and shape of the flame front. The highest concentration of CO_2 corresponds to the flame zone. The low CO_2 concentration region inside the flame corresponds to the pre-heat zone. The low CO_2 concentration region outside the flame corresponds to the post-flame zone. Figure 3 shows the velocity vector field. We can see that the flow is turbulent in the case of lean mixture, when the reaction is relatively slow as well as the flow itself. For the stoichiometric and rich flames the flame speed is higher and the flow becomes laminar.

References

1. Burman, E., Ern, A., Giovangigli, V.: Bunsen flame simulation by finite elements on adaptively refined, unstructured triangulations. Combustion Theory and Modelling 8(1), 65–84 (2003)
2. Consul, R., Pérez-Segarra, C.D., Claramunt, K., Cadafalch, J., Oliva, A.: Detailed numerical simulation of laminar flames by a parallel multiblock algorithm using loosely coupled computers. Combustion Theory and Modelling 7(3), 525–544 (2003)
3. Dunn, M., Masri, A., Bilger, R., Barlow, R.: Finite rate chemistry effects in highly sheared turbulent premixed flames. Flow, Turbulence and Combustion 85(3/4), 621–648 (2010)
4. Fukuchi, N., Takao, J., Hu, C.: Thermal properties and smoke diffusion of oil pool fires in engine room for fire safety design. International Journal of Offshore and Polar Engineering 15(1) (2005)
5. Graziadei, M.: Using local defect correction for laminar flame simulation. Eindhoven University of Technology (2004)
6. van 't Hof, B.: Numerical Aspects of Laminar Flame Simulation. Ph.D. thesis, Eindhoven University of Technology (1998)
7. Kee, R., Rupley, F., Meeks, E., Miller, J.: Chemkin-III: A fortran chemical kinetics package for the analysis of gasphase chemical and plasma kinetics(1996), `http://www.reactiondesign.com/products/open/chemkin.html`
8. Lamorlette, A., Foster, N.: Structural modeling of flames for a production environment. In: Proceedings of the 29th annual conference on Computer graphics and interactive techniques (2002)
9. Mcenally, C., Schaffer, A., Long, M., Pfefferle, L., Smooke, M., Colket, M., Hall, R.: Computational and experimental study of soot formation in a coflow, laminar ethylene diffusion flame. In: Proceedings of the Combustion Institute, pp. 1497–1505 (1998)
10. Nguyen, D., Fedkiw, R., Jensen, H.: Physically based modeling and animation of fire. In: Proceedings of the 29th annual conference on Computer graphics and interactive techniques (2002)
11. Opiola, P.: A new strategy for finite element method adaptation (2010), `http://www.ii.uj.edu.pl/~opiola/adaptive_fem.pdf`
12. Rao, V., Das, P., Sundararajan, T.: An adaptive hp-version of the finite element method applied to flame propagation problems. International Journal for Numerical Methods in Engineering 40(17), 3181–3203 (1997)
13. Skevis, G., Goussis, D., Mastorakos, E.: Understanding methane flame kinetics from reduced mechanisms. International Journal of Alternative Propulsion 1(2/3), 216–227 (2007)
14. Smith, G., Golden, D., Frenklach, M., Moriarty, N., Eiteneer, B., Goldenberg, M., Bowman, C., Hanson, R., Song, S., Gardiner Jr., W.C., Lissianski, V., Qin, Z.: Gri-mech 3.0 (1996), `http://www.me.berkeley.edu/gri_mech`
15. Teigland, R., Eliassen, I.: A multiblock/multilevel mesh refinement procedure for cfd computations. International Journal for Numerical Methods in Fluids 36(5), 519–538 (2001)