A fully coupled, parallel approach for the post-processing of CFD data through reactor network analysis

A. Stagni, A. Cuoci*, A. Frassoldati, T. Faravelli, E. Ranzi

Department of Chemistry, Materials, and Chemical Engineering, Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133 Milano, Italy

Received 12 March 2013
Received in revised form 1 September 2013
Accepted 4 September 2013

1. Introduction

The emissions of pollutant species like nitrogen oxides (NO_x), CO, unburned hydrocarbons and soot particles from combustion devices are regulated from a legislative standpoint in Europe (2010) as well as in the rest of the world. As a result, the design of devices able to reduce the formation of such compounds is vitally important for the concerned industries, and the ability to predict their formation through computational tools with the required accuracy (which is usually of the order of ppm) is then essential.

Such an accuracy can be ensured only by the combination of two factors: an accurate fluid dynamic description of the equipment and a detailed kinetic mechanism, which includes all the elementary chemical reactions involved in the different reaction paths. A full coupling of them, based on a simultaneous resolution of species, momentum and energy equations, is unfeasible because of their mutual dependence and their high degree of non-linearity. This results in an excessive computational load, which can be barely ensured only by modern supercomputers.

In particular, when dealing with turbulent combustion, several assumptions must be made to face the problem: considering a RANS approach (Poinsot & Veynante, 2005) to combustion phenomena (currently the most used in industrial cases), either the turbulent flow or the kinetic mechanism must be simplified in order to reach a reasonable size of the problem and consequently a reasonable computational time (Fichet, Kanniche, Plion, & Gicquel, 2010).

In the last fifteen years, a new family of approaches was conceived, which keeps accuracy on both CFD and kinetics by splitting the resolution procedure into more steps: named post processing techniques, they allow to ease the resolution of the overall numerical problem by exploiting the peculiarities of the system under investigation.

Ehrhardt et al. (1998) first introduced a method of this kind, based on the following key steps:

(1) First, they used a CFD code with a fine grid to evaluate fluid dynamic fields with global kinetic schemes;
(2) Then, through proper algorithms, the computational domain was split into volume elements, whose size was much larger than those of the original CFD cells, thus avoiding an excessive computational load. Each of these volumes was considered as a reactor, which exchanged fixed flows with the neighbors. In this way a network of reactors was created;
(3) Finally, the created reactor network was solved with the assumption of the volume elements being perfectly stirred reactors.

This strategy was then adopted by Frassoldati, Frigerio, Colombo, Inzoli, & Faravelli (2005) after being refined by Faravelli et al. (2001) and Falcitelli, Pasini, Rossi, & Tognotti (2002), who respectively extended their applicability to liquids and proposed a general algorithm to build a reactor network from the CFD field,

* Corresponding author. Tel.: +39 02 2399 3283; fax: +39 02 7063 8173.
E-mail address: alberto.cuoci@polimi.it (A. Cuoci).

Nomenclature

Roman symbols

m	reactor mass (kg)
$\dot{m}$	convective flow (kg/s)
J	diffusion flux (kg/(m^2 s))
S	surface (m^2)
V	reactor volume (m^3)
D	diffusion coefficient (m^2/s)
x	mole fraction (–)
Sc	Schmidt number
N_C	number of reactors
N_S	number of species
t	execution time (s)

Greek symbols

ω	mass fraction (–)
μ	dynamic viscosity (kg/(m s))
ρ	density (kg/m^3)
∇x	gradient of x ($1/x$)

Subscripts

i	species
j	reactor
N_S	turbulent property
s	serial quantity
p	parallel quantity

with positive outcomes on the accuracy of the estimation of pollutants, if compared to experimental data.

A similar approach was introduced by Skjøth-Rasmussen et al. (2004), who solved larger cases without simplifying the computational grid, i.e. considering each computational cell of the CFD grid as a perfectly stirred reactor. They decoupled the material balance from the energy balance by importing and keeping fixed the temperature field from the first step (CFD simulation), thus significantly decreasing the computational load.

A major change in this technique was then introduced by Cuoci, Frassoldati, Buzzi-Ferraris, Faravelli, & Ranzi (2007) and Cuoci et al. (2013), who built up a software, called *Kinetic Post Processor* (KPP), able to post-process CFD data and solve the network through an efficient alternation of local and global solution methods. In particular, they fully exploited the physical structure of the problem in the resolution procedure and translated it numerically into a large, sparse non-linear system of equations. Monaghan et al. (2012) successfully applied this approach to study the pathways of formation of NO and NO_2 in a piloted methane–air diffusion flame. To this purpose, starting from a CFD simulation, they created a reactor network made up of ~1100 perfectly stirred reactors, which was post-processed through the KPP, thus obtaining predictions in very good agreement with experimental data. Van Goethem, Risseeuw, Barendregt, & Frassoldati (2010) effectively exploited the same strategy to study the formation of NO_x in industrial furnaces for the production of ethylene, using this tool also in the design phase, in order to optimize the geometries of the desired equipment.

The performances of this tool were then improved through a deeper focus on the numerical strategy taken by the KPP in order to solve the chemical reactor network created downstream of CFD simulation. An evolution of this tool, named KPPSMOKE (as part of OpenSMOKE libraries (Cuoci, Frassoldati, Faravelli, & Ranzi (2011))), was created, thus being able to manage very large reactor networks (up to 10^6 reactors) with very detailed kinetic mechanisms (hundreds of species and thousands of reactions). Indeed, if not treated properly, those systems may have excessive requirements of computational resources.

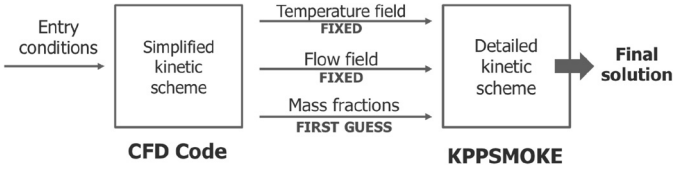

Fig. 1. Kinetic post-processing procedure.

In this paper the development of KPPSMOKE is described in detail by focusing the attention on the two key concepts representing the cornerstones of the newly conceived software: (i) on the one hand, a flexible and efficient strategy to approach the solution was developed, with the help of external libraries for the resolution of the algebraic problems concerned; (ii) on the other, the potentials of the created algorithm were further extended through a parallel distribution of data and tasks over more processes, thus making available a higher memory and computing power.

These two aspects are described in this paper, which is organized as follows: starting from a description of the mathematical model of the reactor network, as well as its constitutive equations (Section 2), the strategy to pursue the solution is explained through an overview of the different techniques adopted (Section 3). Then, Section 4 describes the very core of the work, i.e. the implementation of the numerical model in a parallel code. Its performances are finally benchmarked in Sections 5 and 6, where the comparison of the KPPSMOKE predictions with the experimental data at different scales is presented and its numerical behavior is checked, respectively.

2. Numerical model

The kinetic post-processing approach was developed by relying on one basic assumption: minor species, like radicals and pollutants, do not affect the thermo-fluid dynamic fields (temperature and velocity) in a significant way because of their small amounts (usually ppm). This hypothesis is reasonable as long as they do not have a role in the radiative heat transfer: i.e., while this assumption is largely acceptable for NO_x, CO or polycyclic aromatic hydrocarbons (PAH), it is not for compounds like soot.

Therefore, concerned fluid dynamics can be evaluated upstream through a global, or skeletal, kinetic mechanism, by using one of the available CFD codes for reacting systems (e.g. ANSYS® Fluent (Ansys Inc., 2011), OpenFOAM® (2012), etc.). A fine mesh is usually needed in order to correctly describe the temperature and velocity fields inside the system under investigation. Once this step is complete, the evaluated temperature and velocity fields are exported into the KPPSMOKE and kept fixed during the post-processing analysis (Fig. 1).

The topology of the original computational mesh is imported from the CFD code. Then, a corresponding reactor network model is created in the KPPSMOKE, where each imported cell is assumed as a perfectly stirred reactor, with a fixed temperature and volume. Convective flows among reactors are also obtained in this first step and kept fixed afterwards. These assumptions allow to decouple the mass balances for the species from the energy and momentum balances. Only the former need then to be solved, thus decreasing the overall number of equations and especially their non-linearity (linked to the dependence of kinetic constants from temperature through the Arrhenius' law).

The resulting system is then constituted of $N_C \times N_S$ equations, where N_C and N_S are respectively the number of cells (i.e. reactors) and species, and the unknowns are the related mass fractions. For

the j-th reactor and the i-th species, the steady mass balance can be written in the following way:

$$\dot{m}_j(\omega_{j,i}^{in} - \omega_{j,i}^{out}) + \sum_{k=1}^{N_{AD}}(J_{j,k,i} \cdot S_{j,k}) + V_j \cdot MW_i \cdot R_i = 0 \quad (1)$$

where $\dot{m}_j$ is the convective flow associated to the j-th reactor, $\omega_{j,i}^{in}$ is the average inlet mass fraction of the i-th species in the j-th reactor, $\omega_{j,i}^{out}$ is the mass fraction of the i-th species exiting the j-th reactor, $J_{j,k,i}$ is the diffusion flux of the i-th species across the surface $S_{j,k}$ ($k = 1 \ldots N_{AD}$, i.e. the number of adjacent reactors), V_j the volume of the j-th reactor and MW_i and R_i the molecular weight and the overall formation rate of the i-th species, respectively. Equations' non-linearity is only in the reaction term, as the reaction rates depend on species' concentrations according to the well-known power law. On the other hand, the diffusion fluxes can be written as:

$$J_i = -\left(\rho D_{i,m} + \frac{\mu_t}{Sc_t}\right) \cdot \nabla\omega_i \quad (2)$$

where ρ is the density, $D_{i,m}$ the effective diffusion coefficient of the i-th species, $\nabla\omega_i$ the composition gradient for the i-th species, μ_t the turbulent viscosity and Sc_t the turbulent Schmidt number. In principle, diffusion fluxes are not linear because of $D_{i,m}$, which is a non-linear function of compositions (Wilke, 1950) (while μ_t and Sc_t only depend on the flow field and thus are kept fixed during the post-processing phase). Nevertheless, especially with high Reynolds numbers, the contribution of the $\rho D_{i,m}$ product to (2) is negligible with respect to the turbulent contribution μ_t/Sc_t. Therefore, in the cases concerned, the diffusion flux can be reasonably approximated with the following linear form:

$$J_i = -\frac{\mu_t}{Sc_t} \cdot \nabla\omega_i \quad (3)$$

3. Numerical approach

The use of detailed kinetic schemes makes the system obtained in (1) non-linear and highly stiff. Therefore, segregated approaches like those used by traditional CFD codes cannot be applied to this problem because of the strong interactions between transport and source terms. In this case, the literature shows two possible alternatives to face the issue: (i) multi-operator splitting techniques (Oran & Boris, 2001; Ren & Pope, 2008), which basically solve the governing equations by splitting them into sub-equations, each containing one single operator; (ii) fully coupled algorithms, which solve reaction and transport terms together, with a higher consumption of computational resources, but also a higher degree of accuracy (Kee & Miller, 1977). The former class of methods is less computationally heavy, but also less stable because they separately solve transport and source contributions, and less accurate as they intrinsically introduce a further error (*splitting error*).

The resolution approach was designed considering that the KPPSMOKE is aimed at devices working in steady conditions. In this case, splitting techniques are not always adequate and efficient, because they are generally too slow to face a steady problem (either laminar or turbulent). Actually, Cuoci et al. (2007) showed that a fully coupled strategy, if properly thought, is able to find an accurate steady-state solution. In their steady-state form, mass balances described in (1) form a system of $N_C \times N_S$ non-linear equations (NLS), which may be summarized as:

$$\mathbf{F}(\boldsymbol{\omega}) = 0 \quad (4)$$

where the unknowns are the mass fractions of each species in each reactor. Exploiting the particular structure of the system, an appropriate resolution strategy was developed, which alternates different numerical methods, in order to optimize the approach to the final solution. Each of them is described in the following sections.

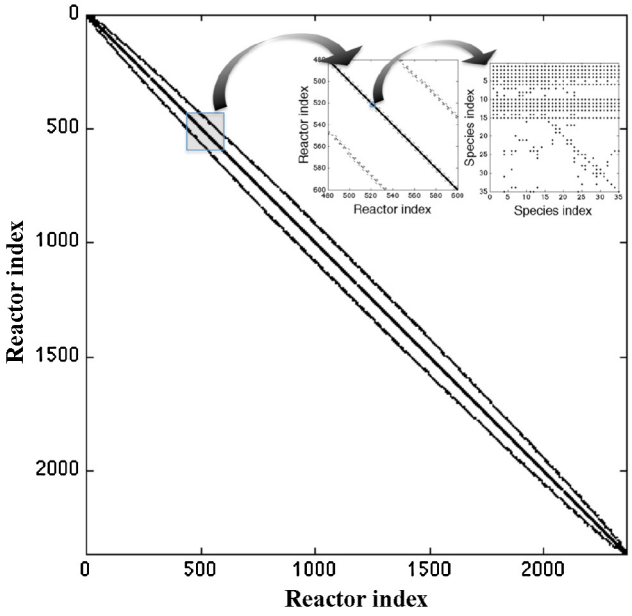

Fig. 2. Jacobian sparsity pattern for a sample reactor network (2360 reactors and 35 species). First enlargement: focus on the non-zero blocks on the main diagonal and next to it. Second enlargement: sparsity pattern of one single block.

3.1. KPPSMOKE as a large and sparse non-linear system

The most natural approach to solve the non-linear system of equations (4) is the use of methods belonging to Newton family: the problem is transformed from a non-linear into a linear, iterative one, through a first-order Taylor expansion in a neighborhood of first guess $\boldsymbol{\omega}_0$ (Deuflhard, 2004):

$$\mathbf{J}(\boldsymbol{\omega}^k)\Delta\boldsymbol{\omega}^k = -\mathbf{F}(\boldsymbol{\omega}^k), \quad \boldsymbol{\omega}^{k+1} = \boldsymbol{\omega}^k + \Delta\boldsymbol{\omega}^k, \quad k = 0, 1, \ldots \quad (5)$$

where $\mathbf{J}(\boldsymbol{\omega}^k)$ is the Jacobian matrix. It is typically very sparse, but generically it is not structured, as shown in Fig. 2.

If equations are ordered cell by cell, the resulting Jacobian matrix is block unstructured, each of which has dimension $N_S \times N_S$. The blocks on the main diagonal are dense, because of the presence of the source terms, while those outside of it are only diagonal sub-matrices because they represent the transport contributions of species of different reactors (which only depend on species gradients, with no interactions among different species because of (3)).

However, the Newton's method is not robust enough to directly solve the non-linear system: the first guess provided by CFD is usually not able to ensure an immediate convergence. Nevertheless, it can be exploited once a better solution is available, as the convergence rate is quadratic if it is sufficiently close to the solution (Deuflhard, 2004).

3.2. Sequential approach

Since in the beginning the problem cannot be solved as a global non-linear system, solution is initially sought by direct substitutions. Under steady-state conditions, the system described in (1) can be viewed in a compact form as:

$$-\mathbf{C}(\boldsymbol{\omega}) + \mathbf{f} + \mathbf{R}(\boldsymbol{\omega}) = \mathbf{0} \quad (6)$$

where the 3 contributions are respectively convection–diffusion, external inlets and reaction terms. A distinction among them can

be made: indeed, if on the one hand convection–diffusion terms depend on the composition of the adjacent cells and of the cell itself, on the other the reaction term only depends on the internal composition of the cell because reactions are a local phenomenon.

Therefore, it is possible to find a local solution by keeping fixed, for each reactor, the inlet transport contributions, evaluated through the composition values of the previous iteration:

$$[\mathbf{C}_{in}(\boldsymbol{\omega}) + \mathbf{f}]_{old} - \mathbf{C}_{out}(\boldsymbol{\omega}) + \mathbf{R}(\boldsymbol{\omega}) = 0 \tag{7}$$

So, instead of having one large $N_C \times N_S$ system, N_C systems of N_S equations can be independently solved. Inlet transport contributions are then updated at the end of each iteration.

Anyway, not always the Newton's method is robust enough to solve even the N_S systems. If not, a local time-stepping method is adopted, with the imposition of a *false transient* (Nauman, 2008), resulting in the following ODE system:

$$\mathbf{m}\frac{d\boldsymbol{\omega}}{dt} = [\mathbf{C}_{in}(\boldsymbol{\omega}) + \mathbf{f}]_{old} - \mathbf{C}_{out}(\boldsymbol{\omega}) + \mathbf{R}(\boldsymbol{\omega}) \tag{8}$$

where $\mathbf{m}$ indicates the total mass of the reactor related to each species. The ODE system reported above is integrated over a time interval sufficient to ensure a significant reduction of local residuals. If they are very low before that threshold, integration automatically stops earlier. A stiff solver is obviously required, because of the presence of species with very different reaction rates. In this way mass fractions can get closer to the solution when no global method would be able to do it. As one could expect, this method progressively loses its effectiveness over the iterations.

3.3. Global resolution through time stepping and Newton method

Once the sequential approach has sufficiently approached the solution, the obtained mass fraction values can then be used as a first guess for a global resolution. First of all, the global approach introduced in Section 3.1 is attempted through a global time stepping, as it was done locally (8) in Section 3.2. The resulting ODE system to be solved is:

$$\mathbf{m}\frac{d(\boldsymbol{\omega})}{dt} = -\mathbf{C}(\boldsymbol{\omega}) + \mathbf{f} + \mathbf{R}(\boldsymbol{\omega}) \tag{9}$$

It is solved using the backward Euler method:

$$\mathbf{m}\frac{\boldsymbol{\omega}^{n+1} - \boldsymbol{\omega}^{n}}{\Delta t} = -\mathbf{C}(\boldsymbol{\omega}^{n+1}) + \mathbf{f} + \mathbf{R}(\boldsymbol{\omega}^{n+1}) \tag{10}$$

A non-linear system is thus obtained, which is linearized like in (5). The length of the time step is adjusted at each iteration, according to the calculated mass fractions: in particular, they are checked to be between 0 and 1, and if these physical constraints are violated, the time step is reduced until the constraints are satisfied. Otherwise, time step is increased with a consequent speed-up in convergence.

If the obtained mass fraction values are near enough to the solution, a significant acceleration in the resolution strategy is provided by the direct use of Newton's method on Eq. (4). It must be remarked that the Jacobian matrix obtained in both resolution via time stepping and Newton's are equal, except for an additive term on the main diagonal in the former, due to the differential term on the left hand side of Eq. (10), which makes the system better conditioned in the first case.

In contrast to its lower robustness, the direct resolution through Newton provides (if successful) a significant acceleration to convergence, thus allowing to reach the final solution of the system in a few iterations.

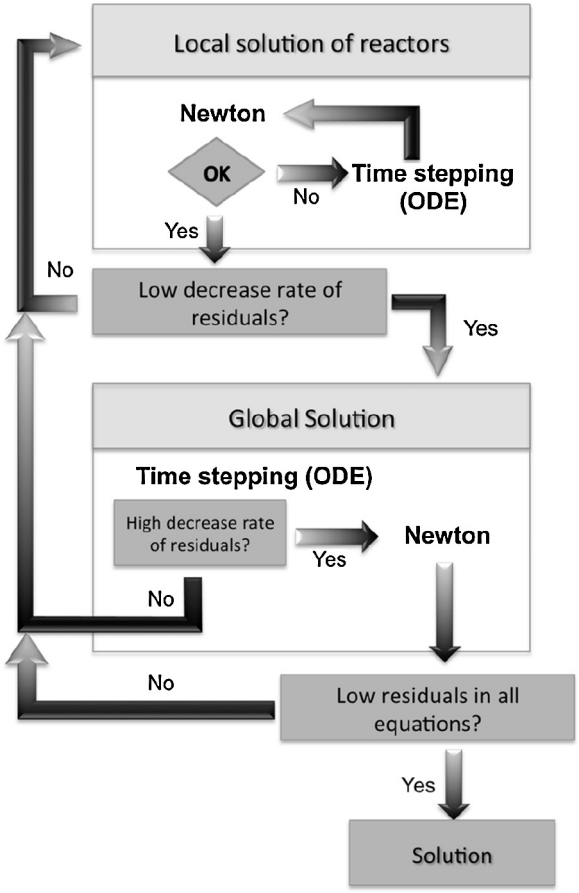

Fig. 3. Procedure for the achievement of convergence (adapted from Cuoci et al. (2007) and Manca, Buzzi-Ferraris, Cuoci, & Frassoldati (2009)).

3.4. Approach to the solution

The methods described above are alternated in order to optimize the approach to the solution. Starting from the first guess provided by the CFD simulation, a sequential approach is applied. This is done with or without time-stepping, according to the positive or negative outcome of local Newton's method. As indicators of convergence, three different norms are accounted for at each iteration by substituting mass fractions into (4): $||\mathbf{F}(\boldsymbol{\omega})||_\infty$, $||\mathbf{F}(\boldsymbol{\omega})||_1$ and $||\mathbf{F}(\boldsymbol{\omega})||_2$, i.e. residuals' infinite norm, norm 1 and norm 2. The overall residuals' trend is checked through norm 1, which is more robust than the others and less sensitive to noise (Ji, 2006). The effectiveness of the sequential method is evaluated through the slope of residuals curve over the iterations: for each iteration, the ratio between the current residuals value and the previous one is evaluated, and the average of these values over the latest n iterations (with n usually equal to 5) is then calculated. If it exceeds the user-entered threshold (e.g. 0.95), a global approach is attempted in the form of global ODE, as described in Section 3.3. In this phase, too, the average residuals ratio is monitored: if it drops below a user-defined threshold (e.g. 0.75), thus meaning that the method is quickly approaching the solution, the global Newton's method is adopted, which significantly accelerates convergence. Otherwise this procedure is repeated, starting from the sequential approach. The whole procedure is graphically summarized in Fig. 3.

4. Algorithm implementation

The KPPSMOKE is part of a collection of libraries named OpenSMOKE (Cuoci et al., 2011), aimed at simulating reacting

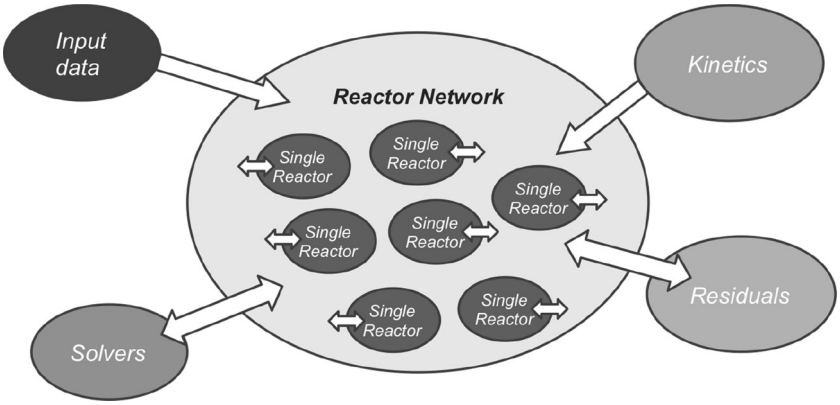

Fig. 4. KPPSMOKE object-oriented structure.

systems with detailed kinetic schemes. Like most of the recent CFD tools, it is written in C++ language, which is based on an Object Oriented Programming (OOP) paradigm. Indeed, such a methodology allows to build an organized (i.e. structured) code in a simpler way: doing so, the whole project can be managed and updated in a more friendly way. Therefore, OOP turns out as particularly useful to model physical problems characterized by the same key elements: species, reactions, kinetic mechanisms and so on. To this purpose, the OpenSMOKE library allows an easy management of the involved reacting mixtures, as its classes are based on a policy design (Alexandrescu, 2001) and proper inheritance techniques.

4.1. Strategy of implementation

In the considered case, the modeled system is made up of the same recurring unit, i.e. the perfectly stirred reactor, whose related objects are then the basic units of the model itself. They receive their operating conditions from two input files imported from the CFD simulation, which respectively fix their first guess' mass fractions and their topology, along with temperature and mass flow rates fields. Reactors exchange data each other through a shared object (the *Reactor Network*), committed to handle the whole grid, i.e. assemble it and solve material balances through the procedure described in Fig. 3. The structure of the algorithm is shown in Fig. 4, where the central role of the *Reactor Network* is pointed out.

The large size of the problems associated with industrial applications made necessary the creation of a structure able to work in distributed memory systems. Most data are stored within reactors: therefore, the modular approach provided by OOP allows an easy distribution over more processes. A static parallelization is carried out by equally splitting the number of total reactors among the available processes, according to the index assigned to each of them by the CFD code. A block-wise distribution among processors was chosen for the reactors: from a physical point of view, it means that the model is block-split in more regions, each of which is assigned to a different process (Fig. 5).

This choice may be disadvantageous in the sequential part of the resolution, where reactors are locally solved. It is possible that the computational load is distributed in a heterogeneous way, e.g. because there are colder regions (i.e. non-reactive) where reactors are usually solved faster than the hotter ones. Therefore, after completing their job some processes may remain idle waiting for the others. On the other hand, this distribution has the advantage that in the global resolution the related linear system is ready to be solved without any need of direct data communication: only the block of the matrix (and right hand side) which pertains to its reactors is assigned to each process, as the adopted linear solver works in a parallel way by splitting the system in blocks.

In order to optimize the computational performances, the adopted parallelization model does not involve any particular hierarchy among processes: a master/slave approach would involve an excessive amount of data passing through the master, and its adoption would then create a bottleneck in the program.

4.2. Local solution

The implementation of such a horizontal parallel algorithm was possible throughout the use of proper libraries of solvers, in both the local and global phases. The former requires the resolution of both differential and non-linear systems whose dimension is N_S (~hundreds of equations in the worst case). Such systems are solved through the *BzzMath* library (Buzzi-Ferraris, 2011), a suite of numerical tools for scientific computing developed at Politecnico di Milano. It constitutes the algebraic skeleton of the program, because of its capability to handle vectors and matrices in a friendly and efficient way. In particular, it is very performing in managing differential stiff systems of equations through the *BzzOdeStiff* class, which adopts the Gear's BDF method to solve them (Buzzi-Ferraris & Manenti, 2012). It checks the error and automatically adjusts the order of the method and the integration step accordingly.

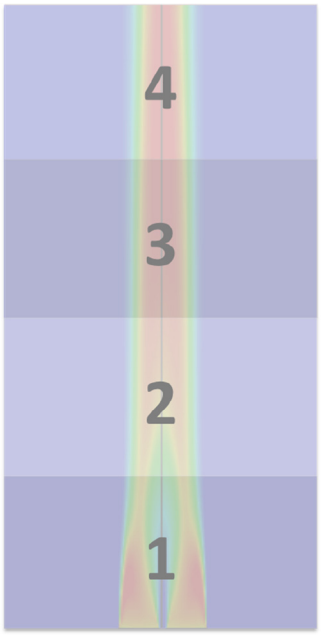

Fig. 5. Example of static distribution of a computational mesh among 4 processors. The sample domain is the HM1 Flame, described in Section 5.1.

Moreover, it is possible to set lower and upper constraints to variables values, which are respectively 0 and 1, as being mass fractions.

Therefore, although the *BzzMath* libraries are not conceived to work in distributed memory systems, they proved suitable to perform such local, independent tasks that do not require any communication. When the latter is needed, e.g. to update mass flow rates after each iteration, the *BzzMath* vectors and matrices are translated into equivalent arrays and vectors provided by the *PETSc* library (Balay et al., 2012), a suite of data structures and routines for the solution of scientific applications modeled by partial differential equations, compatible with the MPI (*Message Passing Interface*) communication standard.

4.3. Global solution

The key issue of this strategy involves the global resolution of the non-linear system described by (1), either directly through Newton's method or by time-stepping (Section 3.3). Both in case of ODE and non-linear systems this problem is iterative, and involves the resolution of the associated linear systems (respectively described in Eqs. (5) and (10)). If a direct solver is used, the related Jacobian matrix needs to be factorized. Proper methods exist to decompose block matrices like the one concerned, but for systems with 10^6 to 10^9 unknowns the time and the needed memory may become unbearable, even for parallel architectures. In particular, as pointed out by Van der Vorst (2003), for these dimensions the use of iterative solvers, less accurate but also less memory and time consuming, becomes the only feasible approach.

As a result, an interface for the iterative solution of the resulting linear systems was created. To this purpose, the *LIS* (*Linear Iterative Solvers*) library (Nishida, 2010) was linked to the program: it is a scalable, parallel software infrastructure written in C language, which gathers a set of system preconditioners and solvers selectable accordingly. Its main advantages consist in linear scalability (the number of iteration counts is $o(n)$ for a problem of size n) and its high parallelization ratio, which, in the latest version, reached a value of 99.99%.

The choice of the proper solver and preconditioner can ensure convergence in a relatively short amount of time. By combining previous experiences reported in the literature (Beers, 2007; Benzi, 2002) with previous test cases on sample grids, it was found that the most performing solving methods belong to Krylov class: in particular, the *BiCGSafe* method (Fujino, Mashami, & Yoshida, 2009) proved to be the most robust and efficient one to solve the linear system, if coupled with an *ILU* (Saad, 2003) or *ILUT* (Saad, 1999) preconditioner.

5. KPPSMOKE validation

The ability of accurately predicting the formation of pollutants through a post processing approach was already assessed in several studies. As an instance, Cuoci et al. (2013) proved its effectiveness in studying several lab-scale turbulent jet flames, fed with different fuels (methane, hydrogen, syngas). On the other hand, the parallel structure allows the KPPSMOKE code to process even larger networks, like those which usually describe industrial equipment.

For this reason, the agreement between the KPPSMOKE predictions and the available experimental data was checked through the assessment of two representative case studies: (i) a hydrogen-methane bluff body flame, known as HM1 (Dally, Masri, Barlow, & Fiechtner, 1998), widely studied in literature as belonging to the TNF flame library (http://www.sandia.gov/TNF), which provides a database of well-documented flames through which combustion models can be validated; (ii) a full-scale, industrial case consisting of a low NO_x, axially staged combustor for aero-engine turbofan, which was already studied by Frassoldati et al. (2009).

5.1. Bluff body flame

As a part of TNF framework, the HM1 flame was experimentally studied at the University of Sidney in several operating conditions. The fuel is fed through a 3.6 mm opening of a 50 mm solid cylinder (Fig. 6), while the coflow air is supplied laterally. In the considered case the fuel velocity is 118 m/s, while the air flows at 40 m/s.

Following the procedure described in Fig. 1, a computational mesh was drawn up according to the indicated geometries. Measured inlet velocity profiles and bluff body surface temperature were set as boundary conditions. The CFD simulation was carried out through the Ansys® Fluent 13.0 software (Ansys Inc., 2011), by adopting a flamelet combustion model. A 2D, axisymmetric, structured mesh (refined in the region close to the bluff-body) made up of 11,979 computational nodes (and 11,760 cells) was adopted. The modified k-ε model was used for the description of turbulence, with a change to the $C_{\varepsilon 1}$ parameter from 1.44 to 1.60, as suggested by McGuirk and Rodi (1979).

The gas mixture was described through a skeletal version of the POLIMI_C1C31201 kinetic scheme (Ranzi et al., 2012), made up of 22 species and 80 reactions.

The results obtained from CFD simulation are reported in Figs. 7 and 8. Clearly, the presence of a toroidal recirculation zone right after the bluff body is predicted (as evident by looking at the velocity fields in Fig. 7c): in this region, residence times are quite higher than those in the rest of the flame, and temperatures are also significantly higher because of the higher conversion.

Looking at the maps of different species (Fig. 8), a direct correspondence can be found between H_2O mass fractions and temperature (Fig. 7a) predictions. CO_2 and CO predictions are coherent with mixture fraction values shown in Fig. 7b, as the latter is formed in the zones richer in fuel, e.g. near the bluff body: here, a lower CO_2 concentration is observed for this reason. Their formation is strictly related to formaldehyde (CH_2O) production, which is an intermediate species, formed as a result of the partial oxidation of fuel (then, it is converted to CO and CO_2 in respectively rich and lean conditions). Last, OH and O maps show the flame front, where CO is oxidized and NO_x are significantly formed through thermal mechanism (Zeldovich, 1946).

A deeper evaluation of these results is provided by their analysis along 6 different traverses. Fig. 9 shows a comparison between the CFD predictions and experimental data in terms of temperature and mixture fraction profiles along the radial direction at several axial locations. The temperature predictions are in good agreement with experimental data for the first two axial locations, while some discrepancies are observed for $x/D \geq 0.90$. In particular, for $x/D = 0.90$ and $x/D = 1.30$, the temperature peak is underestimated, while for the $x/D = 1.80$ and $x/D = 2.40$ it is shifted toward the fuel inlet. On the other hand, the mixture fraction shows a satisfactory trend until $x/D = 1.30$, then some differences from measurements are present.

As explained by Hossain and Malalasekera (2007), the worsening of model predictions after $x/D = 0.90$ are coherent with the known behavior of k-ε model, which usually over-predicts the decay rate of jets. The correction of the $C_{\varepsilon 1}$ coefficient produced an improvement of predictions in the first part of the flame, but discrepancies with experimental data still remained in the farthest coordinates. As a direct consequence, the mixture fraction is under-predicted in the neck zone ($x/D \geq 1.30$), and thus temperature is over-predicted in the neck zone near to the axis.

The CFD solution was kinetically post-processed through the KPPSMOKE code, by applying different detailed kinetic schemes: the GRI 2.11, the GRI 3.0 (Smith et al., 1999), and the POLIMI_C1C31201 (Ranzi et al., 2012) mechanisms. As an example,

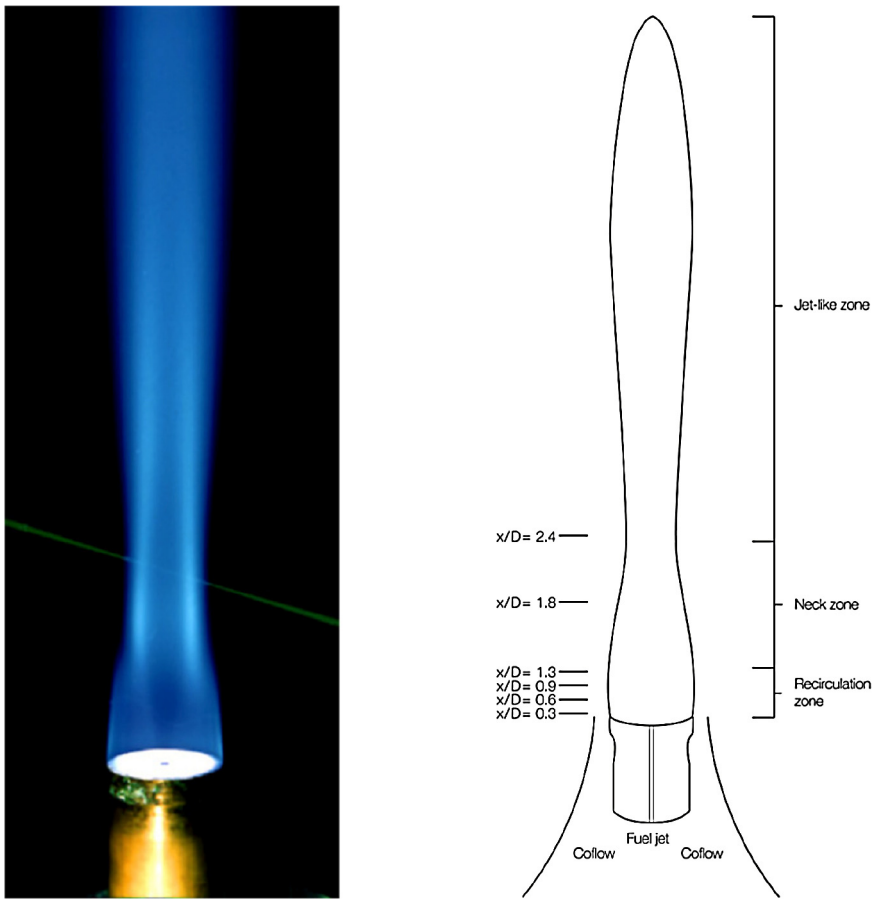

Fig. 6. Left: image of the HM1 experimental set used for the algorithm validation. Right: drawing of the HM1 flame (adapted from Hossain and Malalasekera (2007)), its geometries and main zones (x = longitudinal coordinate; D = Bluff body diameter, i.e. 50 mm).

Fig. 10 shows the maps of mass fractions of CO, NO and HCN obtained through POLIMI_C1C31201 kinetic scheme (111 species and 1835 reactions). The maximum production of NO occurs in the toroidal recirculation zone. Indeed, as pointed out in Fig. 7, it is characterized by high temperatures and residence times, which are the most favorable conditions for thermal NO_x mechanism (which usually prevails over the other mechanisms at high temperatures). On the other hand, HCN is significantly produced in proximity of the bluff body: this zone is rich in fuel (prompt NO_x mechanism (Fenimore, 1971) promotes HCN formation) and with a significant presence of NO, which may result in HCN production because of the reburning mechanism (Frassoldati, Faravelli, & Ranzi, 2003; Glarborg, Østberg, Alzueta, Dam-Johansen, & Miller, 1998).

A detailed analysis of CO and NO predictions was carried out by comparing the profiles along the radial coordinate as estimated by the KPPSMOKE and Fluent with the available experimental data. With regard to CO, Fig. 11 compares its predicted profiles, by both Fluent and KPPSMOKE, to experimental data. With the exception of the last coordinate ($x/D = 2.40$), the POLIMI_C1C31201 mechanism is able to follow the experimental trends; anyway, the variability among the kinetic mechanisms used is averagely low; on the other hand, the variability between Fluent predictions and

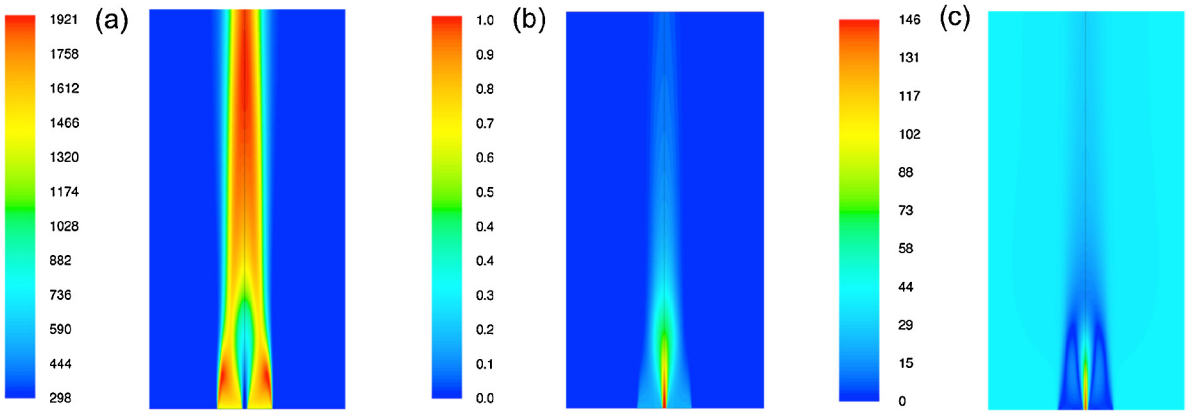

Fig. 7. Bluff-body flame: calculated maps of (a) temperature (K), (b) mixture fraction and (c) velocity magnitude fields. The simulation was performed using the Ansys® Fluent 13.0 CFD code. The size of the computational domain shown is 360 mm × 180 mm.

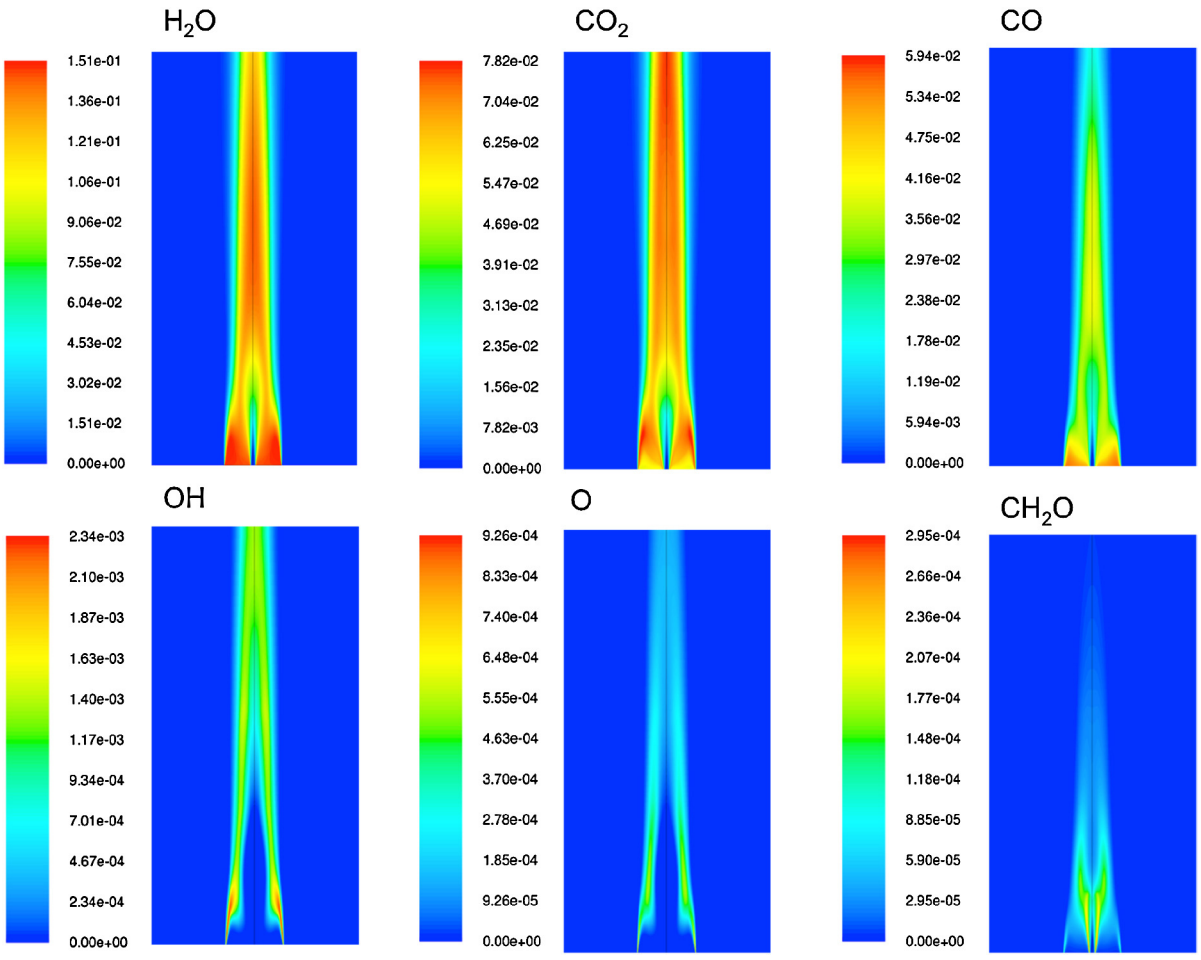

Fig. 8. Calculated maps of mass fractions of several compounds. The size of the reported grid is 360 mm × 180 mm.

KPPSMOKE is more marked because of the different underlying approaches.

The NO profiles along the radial coordinates are reported in Fig. 12 and compared with the experimental data. The POLIMI_C1C31201 mechanism matches experimental data better than others until $x/D = 0.90$; after this threshold, it does not reproduce NO formation peaks, and underestimates its production. GRI 3.0 mechanism over-estimates the NO formation, while better trends are observed for the GRI 2.11 mechanism. The NO over-predictions provided by GRI 3.0 were already observed in literature: Ravikanti, Hossain, & Malalasekera (2009), who developed a laminar flamelet model to predict NO_x formation in the same HM1 flame, experienced a similar output; Sullivan et al. (2002) noticed the same performance in modeling NO_x formation in laminar, ammonia-enriched, methane–air non-premixed flames: they found the cause in the presence of two NH → HNO conversion routes, which leads to a higher presence of HNO over N, and a consequent higher formation of NO. These paths are not present in the earlier GRI 2.11, whose higher reliability for NO_x predictions was extensively shown in literature (Hassan et al., 2013; Kim, Huh, & Dally, 2005).

5.1.1. Comparison with Fluent models

As a further assessment of the proposed approach, the predicted results were compared with the output provided by two Ansys® Fluent 13.0 models able to estimate the formation of NO_x in the same flame: Fluent NO_x processor and the unsteady laminar flamelet model. The Fluent NO_x post-processor solves an additional transport equation involving NO, based on the solution of the other equations (mass, momentum, energy balances) (Ansys Inc., 2009). It is a form of post-processing, although different from the KPPSMOKE: in the case of the Ansys® Fluent model, the NO equation is solved after all the other equations, and therefore it does not have an influence on the other species. In particular, NO evaluation is based on 4 pathways: thermal NO_x, prompt NO_x, fuel NO_x and through N_2O intermediate (if the latter is enabled, a further transport equation involving N_2O is solved). Thermal NO_x is based on the well-known Zeldovich mechanism (Zeldovich, 1946); in the Fluent model, the related kinetic constants are set following the estimations given by Hanson and Salimian (1984). On the other hand, the global kinetic parameter characterizing Prompt NO_x mechanism (proposed for the first time by Fenimore (1971)), is taken from the work of de Soete (1975). Finally, the formation of NO via N_2O intermediate follows the mechanism suggested by Malte and Pratt (1975).

The evaluation of NO was carried out by enabling all the available mechanisms (except the fuel NO_x, which must be considered only in presence of fuel-bound nitrogen). For the thermal model, the instantaneous composition of O and OH was used, while for the N_2O path the *transported-simple* model was adopted. The effect of turbulence was accounted for through a β-PDF distribution on the mixture fraction.

On the other hand, the unsteady laminar flamelet model, implemented in Fluent as well, allows to predict the formation of slow-forming species, which cannot be done with a steady flamelet approach because of the difference in the time-scales between the flamelet time-scale (the inverse of the scalar dissipation) and the time-scales of the species of interest. Indeed, literature (Odedra

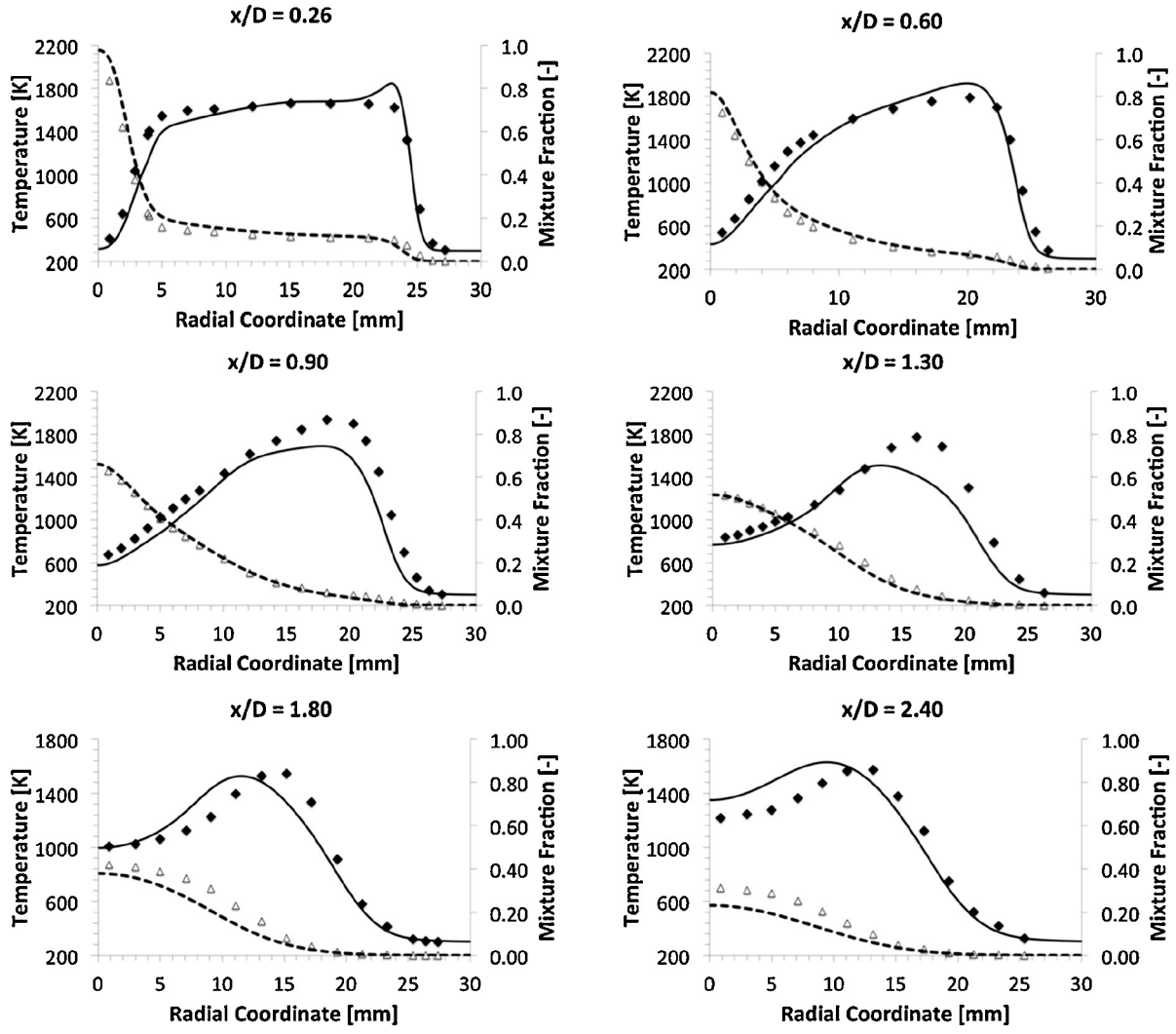

Fig. 9. Bluff-body HM1 flame: predicted vs measured profiles of temperature and mixture fraction along the radial coordinate. Continuous lines: temperature predictions. Dashed lines: mixture fraction predictions. Filled symbols (♦): temperature measurements. Empty symbols (△): mixture fraction measurements.

& Malalasekera, 2007; Pitsch, Chen, & Peters, 1998) shows that a steady flamelet model brings about an overestimation in the formation of NO_x of an order of magnitude, while the output provided by the unsteady approach is more coherent with experimental data.

Fig. 13 shows the different performance between the Ansys Fluent NO_x processor, the KPPSMOKE (via the POLIMI_C1C31201 mechanism) and the Fluent Unsteady Flamelet Model. The main differences are in the lower part of the flame, which is the most critical region of the flame for the NO_x formation, as it is interested by high temperatures, flow recirculations and long residence times. Moreover, in the recirculation region close to the fuel jet local rich conditions are present, where the important role of NO reduction via the reburning mechanism is relevant.

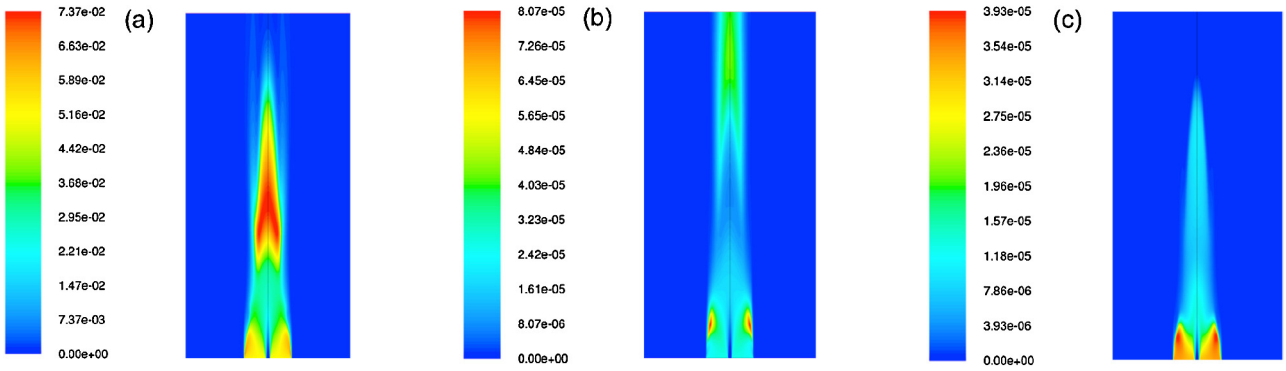

Fig. 10. Bluff-body HM1 flame: calculated maps of (a) CO, (b) NO and (c) HCN mass fractions. The simulation was performed through the KPPSMOKE with POLIMI_C1C31201 kinetic scheme.

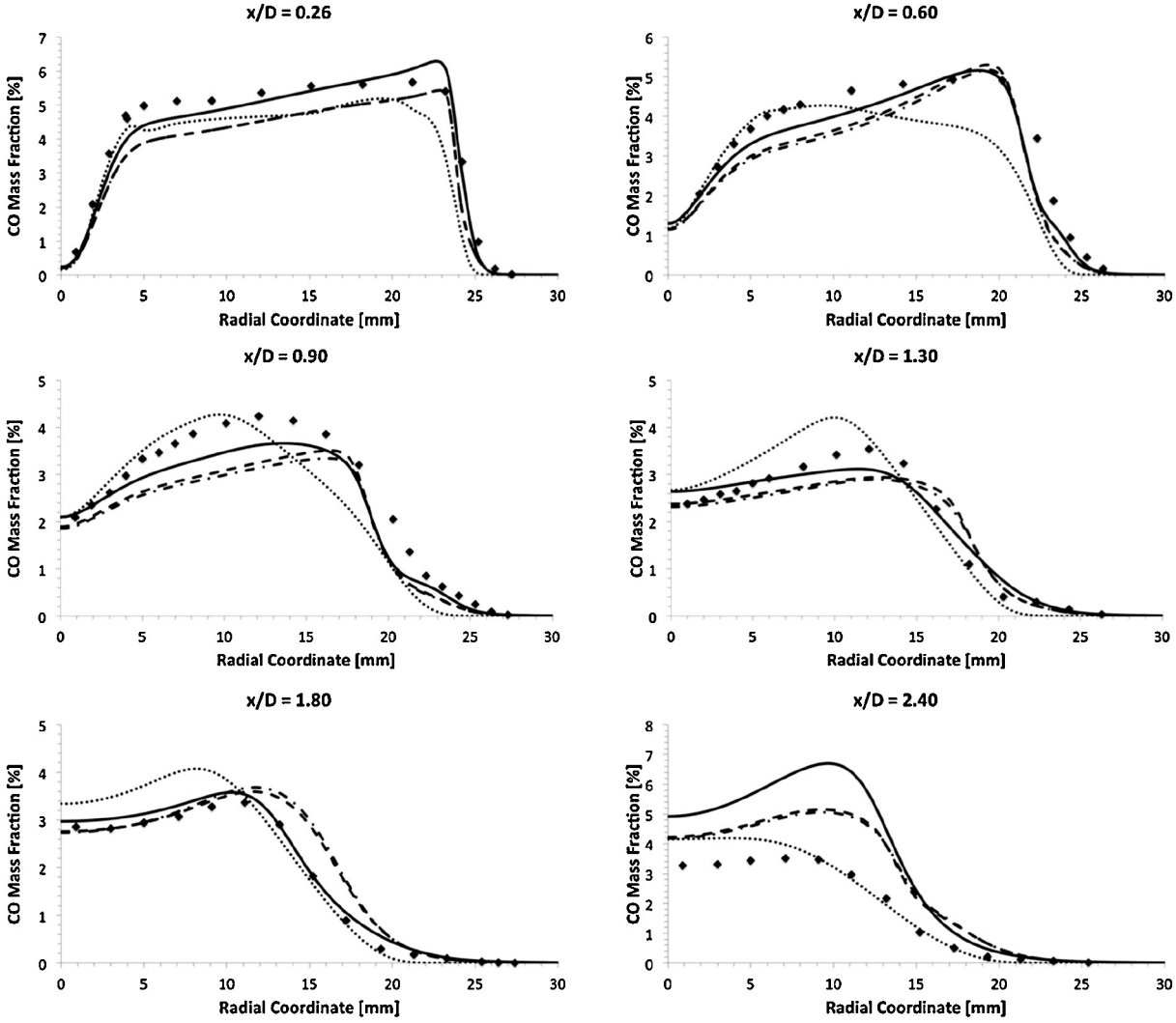

Fig. 11. Bluff-body HM1 flame: predicted (lines) vs measured (symbols) profiles of CO mass fractions. The predictions respectively refer to the Polimi_C1C31201 mechanism (continuous lines), the GRI 2.11 mechanism (dashed lines), the GRI 3.0 mechanism (dash-dot lines) using the KPPSMOKE and Fluent CFD results (dotted lines).

The assessment of the performance of both Fluent models was carried out by comparing them with experimental data, and the related trends are reported in Fig. 14. In all the cases, the NO_x Processor predictions significantly underestimate NO predictions with respect to the KPPSMOKE, which is instead able to better follow experimental trends, although its performance gets worse after *x/D = 1.30*, as already said. Therefore, even if the Ansys® Fluent NO_x processor is computationally much faster (because of the decoupling between NO transport equation from other species equations), the KPPSMOKE output shows a better agreement with experimental data. On the other hand, the predictions obtained through the Unsteady Flamelet model, coupled with POLIMI_C1C31201 mechanism, show a shape more consistent with experimental data, but in all the coordinates it significantly overestimates the measurements.

The worsening of the KPPSMOKE performance from *x/D = 1.30* onwards points out the strong dependence of its output on the accuracy of the CFD simulation used as input: for example, Fig. 9 shows that at *x/D = 1.30* the temperature peak is underestimated of ~300 K, while predicted mixture fraction does not satisfactorily match experimental data in *x/D = 1.80* and *x/D = 2.40*. On the other hand, in the first three coordinates their trend is satisfactory, and so does the output of KPPSMOKE with the POLIMI_C1C31201 mechanism.

5.2. Energy unbalances

The weak influence of the minor species on the temperature fields was checked through an energy analysis of the reactor network. For each reactor, energy unbalances were calculated at the beginning and the end of the resolution procedure:

$$\Delta H_1 = (H_{in} - H_{out})_{start} \tag{11}$$

$$\Delta H_2 = (H_{in} - H_{out})_{end} \tag{12}$$

The overall energy unbalance due to the post-processing was evaluated as the difference of Eqs. (11) and (12). Then, the related unbalance in terms of temperature was calculated as:

$$\Delta T = \frac{\Delta H_2 - \Delta H_1}{\dot{m} c_{p,\mathrm{mix}}} \tag{13}$$

where $\dot{m}$ is the reactor convective flow and $c_{p,\mathrm{mix}}$ is the local specific heat, as evaluated at the local reactor temperature. Fig. 15 shows the local temperature unbalance for HM1 flame, after post-processing it with Polimi_C1C3 mechanism.

Apparently, the temperature unbalance is negligible in almost the whole domain; as one could expect, the most relevant unbalances are obtained in proximity of the flame front, where a difference of some degrees (either positive or negative) can be

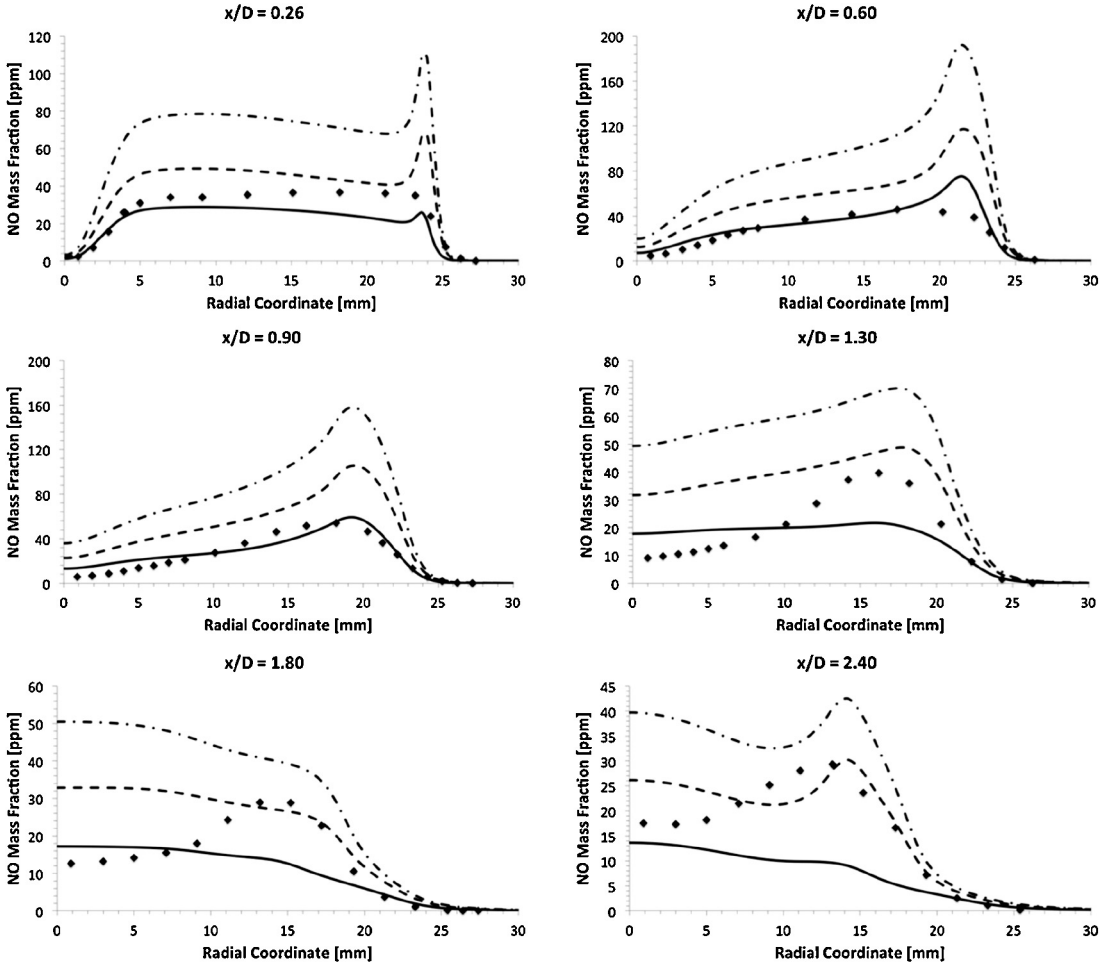

Fig. 12. Bluff-body HM1 flame: predicted (lines) vs measured (symbols) profiles of NO mass fractions. The predictions were obtained through KPPSMOKE, and respectively refer to the Polimi_C1C3 mechanism (continuous lines), the GRI 2.11 mechanism (dashed lines) and the GRI 3.0 mechanism (dash-dot lines).

observed. Indeed, in this region the reaction rates of the main species are higher, and therefore the differences between detailed and skeletal kinetic mechanisms are here emphasized in terms of energy unbalances. Overall, temperature unbalances are relatively small; therefore, these trends confirm the consistency of the results obtained in this case with the basic assumption of the KPPSMOKE indicated in Section 2.

5.3. *CLEAN combustor*

The Low NO_x combustor for aircraft applications described here is an example of industrial equipment interested by turbulent combustion. This study extends what had been carried out by Frassoldati et al. (2009), who applied a serial version of the Kinetic Post Processor, by reducing the reactor network detail through cells

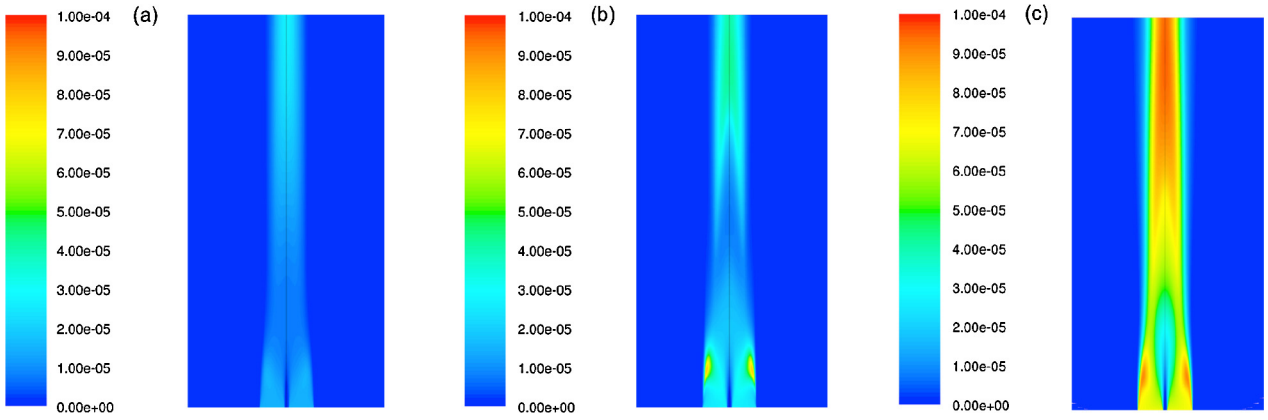

Fig. 13. NO mass fractions maps, as obtained by (a) the Ansys® Fluent NO_x post-processor, (b) the KPPSMOKE (Polimi_C1C3 kinetic mechanism) and (c) the Ansys® Fluent Unsteady Flamelet Model (Polimi_C1C3 kinetic mechanism).

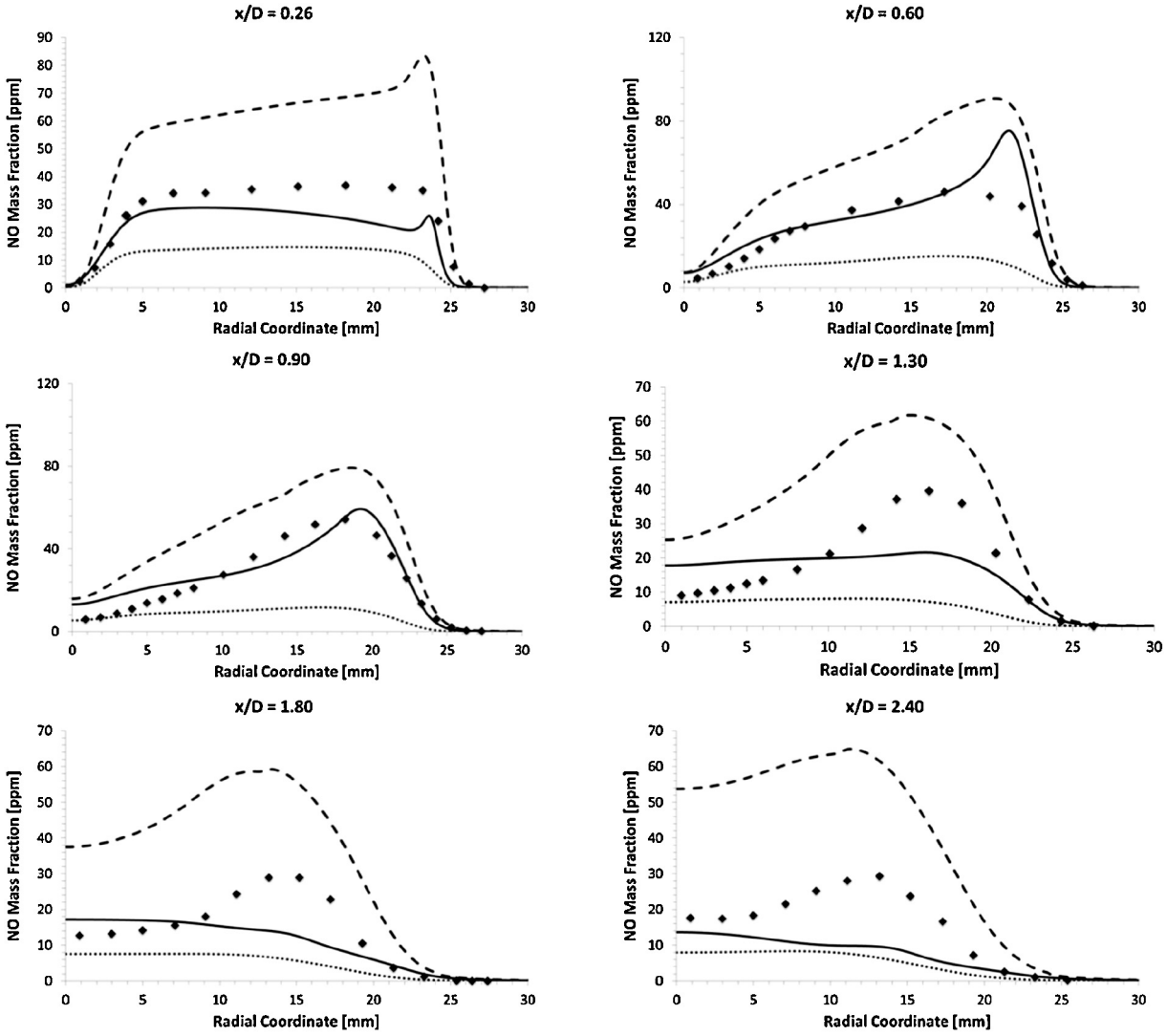

Fig. 14. Bluff-body HM1 flame: predicted (lines) vs measured (symbols) profiles of NO mass fractions. The predictions were obtained through the KPPSMOKE via Polimi_C1C3 mechanism (continuous lines), the Ansys® Fluent NO_x Processor (dotted lines) and the Ansys® Fluent Unsteady Flamelet Model via Polimi_C1C3 mechanism (dashed lines).

clustering (Stagni, Cuoci, Frassoldati, Faravelli, & Ranzi, 2012), necessary for the limited computational resources.

Specifically, the CLEAN combustor is an axially staged combustor equipped with 18 *LPP (Lean Premixed Prevaporised* technology) injectors and 18 conventional pilot injectors (Fig. 16a).

Following the procedure described in Fig. 1, a CFD model of the device (Fig. 16b), made up of 290,764 cells, was created through the BODY3D software developed by AVIO S.P.A. (Di Martino, Cinque, & Terlizzi, 2003). A global two-step kinetic scheme was used for this purpose. The standard k-ε model was used as turbulence model, while the eddy break up concept (Jones & Whitelaw, 1982) was adopted as turbulent combustion model. A two-step global kinetic mechanism was used, made up of the following reactions:

$$C_{12}H_{23} + 11.75O_2 \rightarrow 12CO_2 + 11.5H_2O \quad (14)$$

$$CO + 0.5O_2 \rightarrow CO_2 \quad (15)$$

The CFD code is able to reproduce quite well the fluid dynamic conditions inside the device. As a benchmark, Fig. 17 shows the agreement between CO_2 predictions and experimental data at the combustor exit along the radial coordinate: estimated peaks are slightly lower than measured ones, but overall the two trends are comparable.

The parallel version of the KPPSMOKE code allowed to post-process the obtained CFD data without any reduction in the reactor network detail. For this purpose, the POLIMI_NC7 (86 species and 1427 reactions) kinetic mechanism was used.

Figs. 18 and 19 show a comparison between the measured and calculated mole fraction profiles of NO and CO. It can be observed that the estimated CO profile is in good agreement with experimental data. A significant difference still remains in the evaluation of NO, especially for what concerns the highest levels of emissions. Nevertheless, it must be remarked that the CFD simulation presents some discrepancies with the experimental data: in particular, as shown in Fig. 17, the radial profile of predicted CO_2 does not completely follow the experimental measures. The experimental peaks are higher than the predicted ones. Therefore, it can be deduced that in those regions there is a lower degree of mixing than the one predicted, which results in a higher local temperature, and then a higher production of CO_2, as well as NO.

6. Numerical performances

The effectiveness of the resolution procedure described was tested on several case studies, with different conditions, i.e. in terms of:

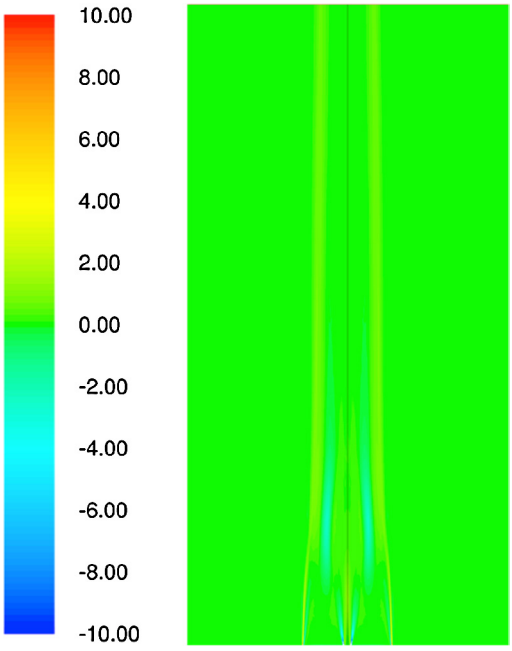

Fig. 15. Local temperature unbalances [K], as obtained by the KPPSMOKE (Polimi_C1C3 kinetic mechanism). The average temperature unbalance is 0.27 K, with a standard deviation of 5.43 K.

– size (i.e. number of equations);
– complexity of the geometry (2D and 3D cases).

In this section the performance of KPPSMOKE for the simulation of 3 different devices is checked through the assessment of the residuals trend over the iterations, and CPU time as a function of the number of processors. Specifically, they represent: (A) a tubular combustor (56,150 cells), discussed by Frassoldati et al. (2010); (B) an aircraft combustor (252,885 cells), which is a further evolution of the work carried out by Frassoldati et al. (2010); (C) a second aircraft combustor (290,764 cells), already described in detail in Section 5.2. All of them were post-processed with the POLIMI_NC7 kinetic mechanism, made up of 86 species and 1427 reactions. Therefore, the resulting non-linear systems are respectively made up of about 4.8 millions, 22 millions and 25 millions of equations.

The simulations described here refer to industrial devices: simpler, lab scale flames are not considered in this analysis because of their easier convergence, lower size and simpler fluid dynamics.

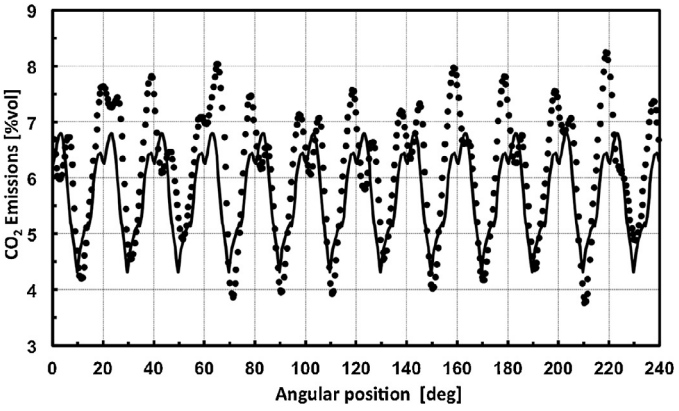

Fig. 17. CLEAN combustor: CO_2 mole fraction profile at combustor exit as predicted by BODY3D (lines) vs. experimental measures (symbols) along the radial coordinate (Frassoldati et al., 2009).

6.1. Residuals behavior

The three simulations used as benchmark respond in different ways to the resolution strategy described in Fig. 3. In particular, Fig. 20 shows the residuals norm 1 as a function of the number of iterations, after normalizing it with respect to their values, as calculated before starting the resolution.

In all the cases, the sequential resolution of the reactor network proves essential in the initial phase to rough out the residuals, although it progressively loses its effectiveness over the iterations. When the global resolution through the time stepping is started, the three systems respond differently: in the first case there is not a significant gain in approaching solution, while the time stepping applied to the other two cases noticeably speeds up convergence rate, as the change in residuals curvature suggests. Finally, the global solution through the Newton's method is able to find the numerical solution in all the cases, as pointed out by the residuals approaching numeric zero in a few iterations.

The most relevant difference in the behavior of these simulations is their response to the time stepping. One of the reasons of this difference is the stiffness degree of the corresponding ODE system which needs to be solved: in spite of the stability ensured by the backward Euler method, too large time steps may result in poor accuracy: indeed, the solution is subject to physical constraints of mass fractions being between 0 and 1. Therefore, if after the integration step any of the unknowns is outside these boundaries, the time step is progressively reduced, with a consequent lower convergence rate. The stiffness degree of the ODE system strongly depends on the kinetic mechanism adopted and fluid dynamic conditions of the flame. As a general rule of thumb, it can be stated that

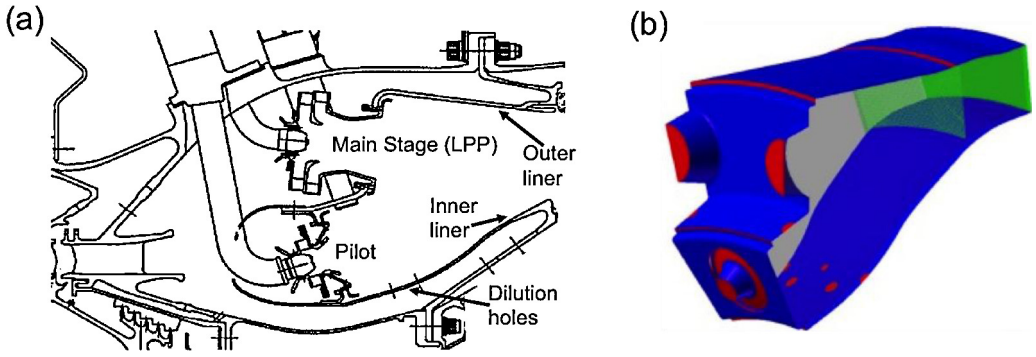

Fig. 16. (a) CLEAN combustor cross-section (Frassoldati et al., 2009). (b) CLEAN combustor: computational domain.

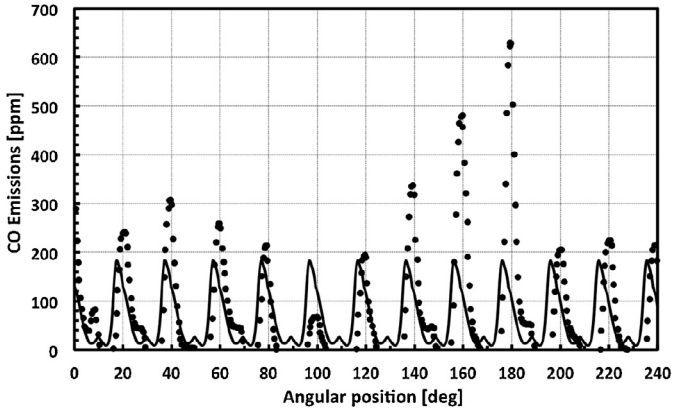

Fig. 18. CLEAN combustor: CO mole fraction profile as predicted by KPPSMOKE (lines) vs. experimental measures (symbols) along the radial coordinate.

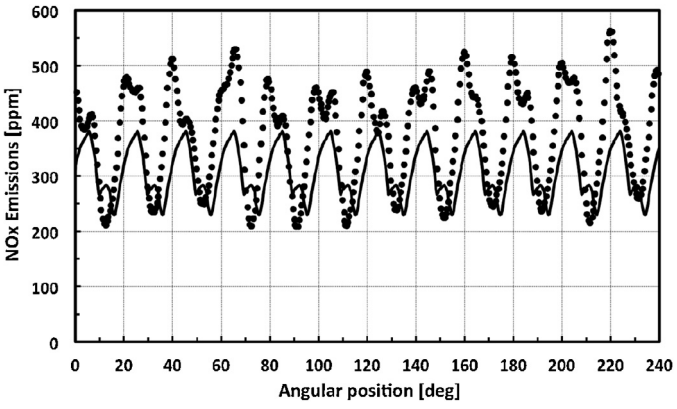

Fig. 19. NO mole fraction profile as predicted by KPPSMOKE (lines) vs. experimental measures (symbols) along the radial coordinate.

the higher the temperature, the faster the kinetics, the stiffer the ODE system.

In terms of computational time, the relative importance of the 3 resolution methods is shown in Fig. 21. Although these percentages are strongly dependent on the specific case, most of the time is spent by the sequential resolution (∼50%). Anyway, the relative weight of the three methods can be properly tuned by adjusting the program options, which allow to set the relative and absolute

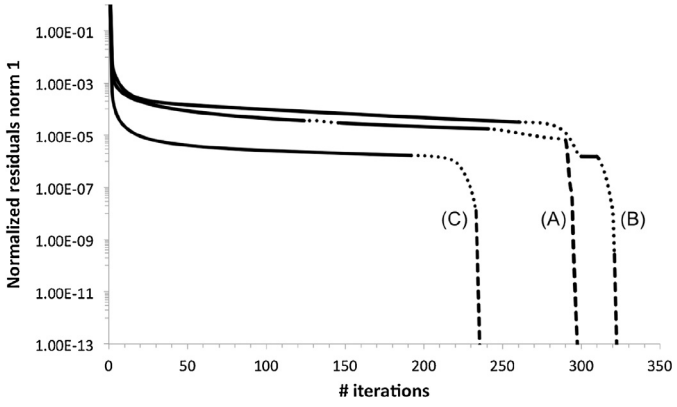

Fig. 20. Residuals norm 1 trends, normalized with respect to their initial value (set equal to 1). Continuous lines: resolution through the sequential approach. Dotted lines: resolution through the global time stepping. Dashed lines: resolution as a global non-linear system.

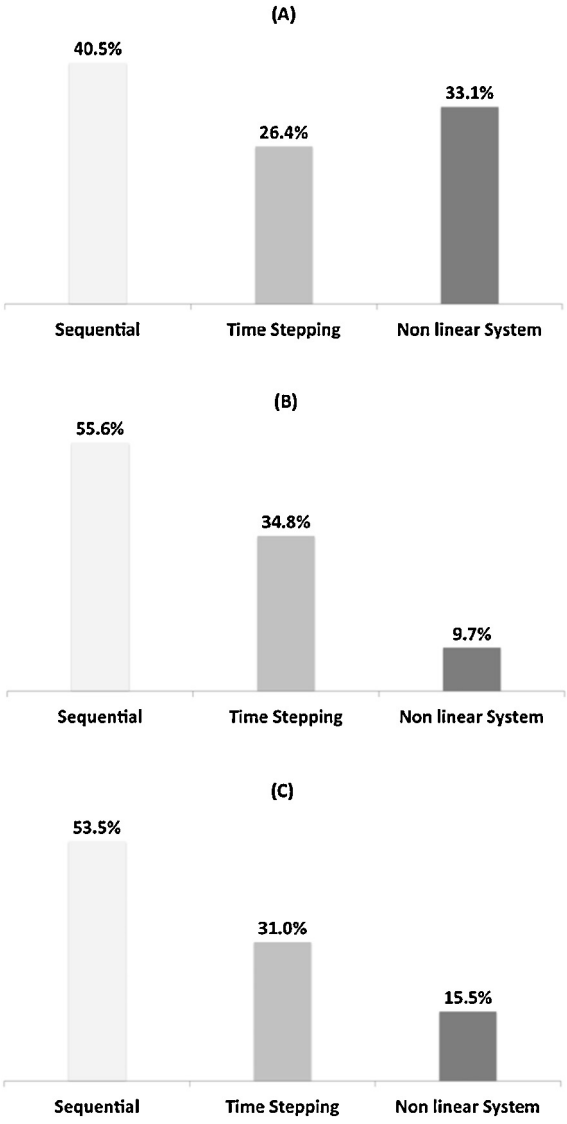

Fig. 21. Relative importance of the 3 resolution methods in terms of time. Dark gray: resolution as a global non-linear system. Medium gray: resolution through global time stepping. Light gray: resolution through sequential approach. This benchmark was carried out using 48 processors.

tolerance of each of the approaches, as well as the switch criteria between them, described in Section 3.4.

6.2. *Time performance and scalability*

The numerical efficiency of KPPSMOKE was investigated in terms of execution time of the algorithm on multiprocessor architectures. For this purpose, a Linux cluster was used, made up of 16 nodes connected through Infiniband technology with each 2 Intel Xeon X5675 (3.07 GHz) processors, each having 6 cores.

Of the three full-scale cases used as benchmark, only the first one could be executed serially for memory reasons: indeed, the required amount of memory in the others exceeded the available RAM of the single node. Therefore, case B was tested starting from 7 processors, while case C from 8 processors (both with 1 per node).

In Fig. 22 the CPU times required to the three test cases to achieve convergence, with a variable number of processors, are plotted. For case A, the serial execution requires about 49 h to reach the convergence. With more processors, time is progressively reduced up to ∼50 min. Similar decrease rates are observed for cases B and C.

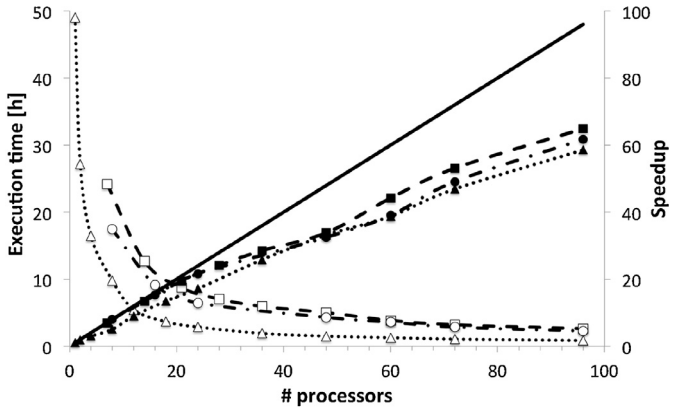

Fig. 22. Time performance of the three test flames. Dotted line with empty triangles (△): execution time of case A. Dashed line with squares (□): execution time of case B. Dash-dotted line with points (○): execution time of case B. Continuous line: ideal speedup. Dotted line with filled triangles (▲): speedup of case A. Dashed line with squares (■): speedup of case B. Dash-dotted line with filled points (●): speedup of case C.

Fig. 22 also shows the results in terms of attained speedup S_p, evaluated as:

$$S_p = \frac{t_s}{t_p} \tag{16}$$

where t_s and t_p are respectively the execution time with one processor and execution time with p processors. In those cases where simulation could not be run with one processor (cases B and C), it is evaluated as:

$$S_p = \frac{t_{p_{\min}}}{t_p} \cdot p_{\min} \tag{17}$$

where $p_{\min}$ is the minimum number of processors used to run the related simulation, and $t_{p_{\min}}$ the associated execution time.

As it can be noted, the combined effect of load unbalancing and communication overhead results in an efficiency loss of 35–40% when about 100 processors are used. This effect is felt in a similar way in all of the three flames tested, and is unavoidably related to the physical structure of the problem as well as to the iterative approach developed and the block-wise distribution of the reactor network among nodes and processors.

7. Conclusions and future developments

The implementation of post processing techniques for the estimation of pollutants from industrial devices is a challenging task for several reasons: the numerical complexity of the resulting problem, as well as its high CPU time and memory requirements, which make unavoidable the use of parallel computing systems. On the other hand, the benefits in the prediction of emissions from related equipment are apparent, as widely shown in literature.

In this context, KPPSMOKE was conceived. By exploiting the negligible influence of minor species on temperature and velocity fields, the estimation of pollutants formation is carried out in 2 main steps: (i) fluid dynamic fields are evaluated through one of the available CFD codes, by adopting a global or a skeletal kinetic mechanism; (ii) then, a network of perfectly stirred reactors is created, whose geometries and fluid dynamic conditions (temperature and velocity) are imported from the first step. In this way a large, block sparse non-linear system of $N_C \times N_S$ mass balance equations is obtained, which is solved through a proper numerical procedure. It alternates the use of local and global solution methods in order to reach the solution of the problem in an efficient, flexible and robust way. External libraries like *BzzMath*, *LIS* and *PETSc* were used in the algorithm, so as to create an optimized parallel code.

The developed algorithm shows all the advantages and limitations of a RANS-based approach: it allows to create and manage models of full-size devices with an acceptable computing load, such that it can be used also outside the academic environment for industrial purposes. Nevertheless, in RANS approaches the equations are averaged in time over all turbulent scales and therefore rely on models to evaluate the effects of turbulent fluctuations. On the other hand, more computationally expensive methods like LES-based provide substantial advantages for modeling turbulent combustion, especially due to the better description of the large-scale motion of the turbulence and the scalar mixing process.

Both these aspects were proven by testing KPPSMOKE under two different points of view. First of all, the algorithm was validated on representative test cases, of both lab and industrial scale. The numerical results confirmed the effectiveness of this tool, but they also pointed out a strong dependence of the predictions on the accuracy of CFD input data and on the detailed kinetic mechanism used for the post-processing. On the other hand, from a numerical standpoint, the overall performance of the algorithm is quite satisfactory, since it proved able to solve cases of industrial interest in acceptable times (∼hours), when a parallel infrastructure is used.

The validation of the KPPSMOKE was carried out in terms of NO and CO, but in principle its range of applicability can be extended to more complex chemical species: as outlined in the beginning, the tool can be applied to all those pollutants that do not have a significant influence on the thermal field. Therefore, starting from the available tool, the prediction of the formation of unburned hydrocarbons and polycyclic aromatic hydrocarbons (PAH) will also be an important topic to focus on in the future. In order to reach this goal, the use of more complex (and larger) detailed kinetic mechanisms than those used for NO becomes necessary (more than 200 species), with a consequent increase in the requirements of computational resources.

On the other hand, memory and power consumption can still be optimized through two further improvements in the current algorithm: (i) replacement of the current linear iterative solving method (Sections 3.3 and 4.3) with a more suitable version for the considered system, whose block structure is not currently fully exploited by the general solver for sparse, asymmetric systems (*LIS*) used in this work. The implementation of a block Jacobi method (Quarteroni et al., 2007) could be the next evolution in this direction, due to its intrinsic high parallelizability; (ii) implementation of the *nested parallelism* model in order to exploit at best the features of the modern parallel architectures, made up of hybrid distributed-shared memory systems. This can be done by implementing an OpenMP interface (www.openmp.org) in addition to the MPI interface, that can bring about a considerable saving in memory consumption, as well as possible improvements in execution time and attainable speedup.

Acknowledgments

This work was carried out through the financial support of the European Union (EU) as part of the EMICOPTER Project (CS-GA-2009-1261 251798). The authors would like to acknowledge the experimental and computational support provided by AVIO S.P.A., and the active collaboration of Carlo Maria Campelli, Carlo De Falco and Luca Formaggia (Politecnico di Milano) for the development of KPPSMOKE. The support provided by Akira Nishida (Kyushu University) and Marco van Goethem (Technip Benelux B.V.) is also fully recognized.

References

Alexandrescu, A. (2001). *Modern C++ design: Generic programming and design patterns applied. C++ in-depth series*. Boston, MA: Addison-Wesley.

Ansys Inc. (2009). *Fluent 12 user's guide*.

Ansys Inc. (2011). *ANSYS® Fluent, Release 13.0, User's Guide*. ANSYS Inc.

Balay, S., Brown, J., Buschelman, K., Gropp, W. D., Kaushik, D., Knepley, M., et al. (2012). *PETSc web page.* , available from: http://www.mcs.anl.gov/petsc

Beers, K. J. (2007). *Numerical methods for chemical engineering: Applications in Matlab*. Cambridge, UK; New York: Cambridge University Press.

Benzi, M. (2002). Preconditioning techniques for large linear systems: A survey. *Journal of Computational Physics*, *182*(2), 418–477.

Buzzi-Ferraris, G. (2011). *BzzMath: Numerical libraries in C++.* , available from: www.chem.polimi.it/homes/gbuzzi

Buzzi-Ferraris, G., & Manenti, F. (2012). BzzMath: Library overview and recent advances in numerical methods. *Computer Aided Chemical Engineering*, *30*(2), 1312–1316.

Cuoci, A., Frassoldati, A., Buzzi-Ferraris, G., Faravelli, T., & Ranzi, E. (2007). The ignition, combustion and flame structure of carbon monoxide/hydrogen mixtures. Note 2: Fluid dynamics and kinetic aspects of syngas combustion. *International Journal of Hydrogen Energy*, *32*(15), 3486–3500.

Cuoci, A., Frassoldati, A., Faravelli, T., & Ranzi, E. (2011). *OpenSMOKE: Numerical modeling of reacting systems with detailed kinetic mechanisms*. Rome: XXXIV Meeting of the Italian Section of the Combustion Institute.

Cuoci, A., Frassoldati, A., Stagni, A., Faravelli, T., Ranzi, E., & Buzzi-Ferraris, G. (2013). Numerical modeling of NO_x formation in turbulent flames using a kinetic post-processing technique. *Energy and Fuels*, *27*(2), 1104–1122.

Dally, B. B., Masri, A. R., Barlow, R. S., & Fiechtner, G. J. (1998). Instantaneous and mean compositional structure of bluff-body stabilized nonpremixed flames. *Combustion and Flame*, *114*(1–2), 119–148.

de Soete, G. G. (1975). Overall reaction rates of NO and N_2 formation from fuel nitrogen. *Symposium (International) on Combustion*, *15*(1), 1093–1102.

Deuflhard, P. (2004). *Newton methods for nonlinear problems: Affine invariance and adaptive algorithms. Springer series in computational mathematics*. Berlin; New York: Springer.

Di Martino, P., Cinque, G., & Terlizzi, A. (2003). Pilot fuel injection optimization in an annular combustor. In *9th international conference on liquid atomization and spray systems* Sorrento, Italy.

Ehrhardt, K., Toqan, M., Jansohn, P., Teare, J. D., Beer, J. M., Sybon, G., et al. (1998). Modeling of NO_x reburning in a pilot scale furnace using detailed reaction kinetics. *Combustion Science and Technology*, *131*(1–6), 131–146.

European Parliament C. (2010). Directive 2010/75/EU of the European Parliament and of the council of 24 November 2010 on industrial emissions (integrated pollution prevention and control). *Official Journal of the European Union*.

Falcitelli, M., Pasini, S., Rossi, N., & Tognotti, L. (2002). CFD plus reactor network analysis: An integrated methodology for the modeling and optimisation of industrial systems for energy saving and pollution reduction. *Applied Thermal Engineering*, *22*(8), 971–979.

Faravelli, T., Bua, L., Frassoldati, A., Antifora, A., Tognotti, L., & Ranzi, E. (2001). A new procedure for predicting NO(x) emissions from furnaces. *Computers and Chemical Engineering*, *25*(4–6), 613–618.

Fenimore, C. P. (1971). Formation of nitric oxide in premixed hydrocarbon flames. *Symposium (International) on Combustion*, *13*(1), 373–380.

Fichet, V., Kanniche, M., Plion, P., & Gicquel, O. (2010). A reactor network model for predicting NO_x emissions in gas turbines. *Fuel*, *89*(9), 2202–2210.

Frassoldati, A., Faravelli, T., & Ranzi, E. (2003). Kinetic modeling of the interactions between NO and hydrocarbons at high temperature. *Combustion and Flame*, *135*(1–2), 97–112.

Frassoldati, A., Frigerio, S., Colombo, E., Inzoli, F., & Faravelli, T. (2005). Determination of NO_x emissions from strong swirling confined flames with an integrated CFD-based procedure. *Chemical Engineering Science*, *60*(11), 2851–2869.

Frassoldati, A., Cuoci, A., Faravelli, T., Ranzi, E., Colantuoni, S., Martino, P. D., et al. (2009). Experimental and modeling study of a low NO_x combustor for aero-engine turbofan. *Combustion Science and Technology*, *181*(3), 483–495.

Frassoldati, A., Cuoci, A., Faravelli, T., Ranzi, E., Colantuoni, S., Di Martino, P., et al. (2010). Fluid dynamics and detailed kinetic modeling of pollutant emissions from lean combustion systems. *Proceedings of the ASME Turbo Expo*, *2*(Pts A and B), 451–459.

Fujino, S., Mashami, M., & Yoshida, M. (2009). A proposal of preconditioned BiCGSafe method with safe convergence. *Computers and Chemical Engineering*, *33*, 1727–1734.

Glarborg, P., Østberg, M., Alzueta, M. U., Dam-Johansen, K., & Miller, J. A. (1998). The recombination of hydrogen atoms with nitric oxide at high temperatures. *Symposium (International) on Combustion*, *27*(1), 219–226.

Hanson, R., & Salimian, S. (1984). Survey of rate constants in the N/H/O system. In W. C. Gardiner Jr. (Ed.), *Combustion chemistry* (pp. 361–421). USA: Springer.

Hassan, G., Pourkashanian, M., Ingham, D., Ma, L., Newman, P., & Odedra, A. (2013). Predictions of CO and NO_x emissions from steam cracking furnaces using GRI2.11 detailed reaction mechanism – A CFD investigation. *Computers and Chemical Engineering*, *58*(0), 68–83.

Hossain, M., & Malalasekera, W. (2007). A combustion model sensitivity study for CH_4/H_2 bluff-body stabilized flame. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *221*(11), 1377–1390.

Ji, J. (2006). CGG method for robust inversion and its application to velocity-stack inversion. *Geophysics*, *71*(4), R59–R67.

Jones, W. P., & Whitelaw, J. H. (1982). Calculation methods for reacting turbulent flows: A review. *Combustion and Flame*, *48*(0), 1–26.

Kee, R. J., & Miller, J. A. (1977). *Split-operator, finite difference solution for axisymmetric laminar jet diffusion flames*. Albuquerque, NM, USA: Sandia Laboratories.

Kim, S. H., Huh, K. Y., & Dally, B. (2005). Conditional moment closure modeling of turbulent nonpremixed combustion in diluted hot coflow. *Proceedings of the Combustion Institute*, *30*, 751–757.

Malte, P. C., & Pratt, D. T. (1975). Measurement of atomic oxygen and nitrogen oxides in jet-stirred combustion. *Symposium (International) on Combustion*, *15*(1), 1061–1070.

Manca, D., Buzzi-Ferraris, G., Cuoci, A., & Frassoldati, A. (2009). The solution of very large non-linear algebraic systems. *Computers and Chemical Engineering*, *33*(10), 1727–1734.

McGuirk, J. J., & Rodi, W. (1979). The calculation of three-dimensional turbulent free jets. In *1st symposium on turbulent shear flows*.

Monaghan, R. F. D., Tahir, R., Cuoci, A., Bourque, G., Furi, M., Gordon, R. L., et al. (2012). Detailed multi-dimensional study of pollutant formation in a methane diffusion flame (Vol. 26, p. 1598, 2012). *Energy and Fuels*, *26*(4), 2546.

Nauman, E. B. (2008). *Chemical reactor design, optimization, and scaleup* (2nd ed.). Hoboken, NJ: Wiley.

Nishida, A. (2010). Experience in developing an open source scalable software infrastructure in Japan. *Proceedings of Computational Science and Its Applications – Iccsa 2010, Pt 2*, *6017*, 448–462.

Odedra, A., & Malalasekera, W. (2007). Eulerian particle flamelet modeling of a bluff-body CH_4/H-2 flame. *Combustion and Flame*, *151*(3), 512–531.

OpenFOAM Foundation. (2012). *OpenFOAM User Guide, Version 2.1.1*.

Oran, E. S., & Boris, J. P. (2001). *Numerical simulation of reactive flow* (2nd ed.). Cambridge, UK; New York, NY: Cambridge University Press.

Pitsch, H., Chen, M., & Peters, N. (1998). Unsteady flamelet modeling of turbulent hydrogen-air diffusion flames. In *Twenty-seventh symposium (international) on combustion, Vols. 1 and 2* (pp. 1057–1064).

Poinsot, T., & Veynante, D. (2005). *Theoretical and numerical combustion* (2nd ed.). Philadelphia: Edwards.

Quarteroni, A., Sacco, R., & Saleri, F. (2007). Numerical mathematics. In *Texts in applied mathematics* (2nd ed.). Berlin; New York: Springer.

Ranzi, E., Frassoldati, A., Grana, R., Cuoci, A., Faravelli, T., Kelley, A. P., et al. (2012). Hierarchical and comparative kinetic modeling of laminar flame speeds of hydrocarbon and oxygenated fuels. *Progress in Energy and Combustion*, *38*(4), 468–501.

Ravikanti, M., Hossain, M., & Malalasekera, W. (2009). Laminar flamelet model prediction of NO_x formation in a turbulent bluff-body combustor. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, *223*(1), 41–54.

Ren, Z. Y., & Pope, S. B. (2008). Second-order splitting schemes for a class of reactive systems. *Journal of Computational Physics*, *227*(17), 8165–8176.

Saad, Y. (1999). ILUT: A dual threshold incomplete LU factorization. *Numerical Linear Algebra with applications*, *1*(4), 387–402.

Saad, Y. (2003). *Iterative methods for sparse linear systems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Skjøth-Rasmussen, M. S., Holm-Christensen, O., Østberg, M., Christensen, T. S., Johannessen, T., Jensen, A. D., et al. (2004). Post-processing of detailed chemical kinetic mechanisms onto CFD simulations. *Computers and Chemical Engineering*, *28*(11), 2351–2361.

Smith, G. P., Golden, D. M., Frenklach, M., Moriarty, N. W., Eiteneer, B., Goldenberg, M., et al. (1999). *GRI 3.0 kinetic mechanism.* , available from: http://www.me.berkeley.edu/gri_mech/

Stagni, A., Cuoci, A., Frassoldati, A., Faravelli, T., & Ranzi, E. (2012). Effect of clustering on reactor network models. In *35th meeting of the Italian section of the Combustion Institute* Milano.

Sullivan, N., Jensen, A., Glarborg, P., Day, M. S., Grcar, J. F., & Bell, J. B. (2002). Ammonia conversion and NOX formation in laminar coflowing nonpremixed methane–air flames. *Combustion and Flame*, *131*(3), 285–298.

Van der Vorst, H. A. (2003). *Iterative Krylov methods for large linear systems*. Cambridge, UK: Cambridge University Press.

Van Goethem, M., Risseeuw, I., Barendregt, S., & Frassoldati, A. (2010). The design of ultra-low NOx critical furnaces. In *AIChE annual meeting* San Antonio, TX.

Wilke, C. R. (1950). Diffusional properties of multicomponent gases. *Chemical Engineering Progress*, *46*, 95–104.

Zeldovich, Y. B. (1946). The oxidation of nitrogen in combustion and explosions. *Acta Physicochemica, USSR*, *21*, 577–628.