Large Eddy Simulation of combustion on massively parallel machines

Gabriel Staffelbach[1], Jean Mathieu Senoner[1], Laurent Gicquel[1] and Thierry Poinsot[2]

[1] CERFACS, 42 avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France
[2] IMFT (UMR CNRS-INPT-UPS 5502), Toulouse, France

Abstract. Combustion is the source of eighty percent of the energy produced in the world: it is therefore a topic of major interest in the present context of global warming and decreasing fuel resources. Simulating combustors and especially instability mechanisms in these systems has become a central issue in the combustion community in the last ten years. This can be achieved only on massively parallel computers. This paper presents modern techniques to simulate reacting flows in realistic geometries and shows how parallel computing (typically thousands of processors) has made these simulations possible. The physics of reacting flows are only discussed briefly to concentrate on specific issues linked to massively parallel computing, to the turbulent character of the flow and the effects of rounding errors.

1 Introduction

This paper presents an overview of combustion and of CFD (Computational Fluid Dynamics) for combustion. It focuses on the place of instabilities in reacting flows and on the role of massively parallel computations. These instabilities are found at many levels:

- Like any shear flow, reacting flows are submitted to hydrodynamic instabilities [1, 2] and to vortex formation. Such vortices are easily observed in nature, like in the wake of aircrafts for example. When they are found within combustion chambers, they can constitute a major danger.
- Acoustics play a major role in reacting flows: by coupling with heat release, they are the source of a major problem in many combustion devices: combustion instabilities [3, 4] which can induce high vibration levels and, in extreme cases, destroy combustion hardware in a few seconds.
- Instabilities are present in the physical problem to study but they are also present in the numerical methods used to simulate these mechanisms. Most high-fidelity numerical schemes required for Computational Fluid Dynamics exhibit low dissipation and therefore multiple non-physical instabilities (wiggles) which can require significant efforts to be kept under control [5, 6, 4].
- Finally, CFD for reacting flows is performed today on massively parallel machines: these architectures coupled with centered schemes needed for turbulent flows lead to an additional type of instability linked to the growth of rounding errors and to a new type of

instability where the solution depends on unexpected parameters such as the commutativity errors of addition, the initial condition or the number of processors.

All these phenomena are 'instabilities' even though they correspond to very different physical mechanisms. In many cases, they can couple and in LES of combustion instabilities, the first issue is to be able to control the non physical waves due to the high-order spatial scheme as well as the rounding errors due to massively parallel computing. In this paper, the physics of combustion and of instabilities are briefly discussed before presenting a code used for LES of combustion by multiple groups and discussing one specific issue linked to the effect of rounding errors in simulations of turbulent flows.

2 Combustion: the source of our energy

Combustion is the unknown heart of most present problems discussed everyday on global change and pollution issues. More than eighty percent of the energy produced on earth is obtained by burning some fossil fuel. This combustion can be produced by burning wood and producing a few Watts or by running 20 meter long industrial turbines producing 200 MWatts or more. The processes used for combustion can be simplified and non optimized like for wood combustion or highly technological like in reciprocating engines. This makes combustion the first contributor to our life style, our energy consumption and to the production of pollutants such as NO_x or CO_2. This also implies that controlling global change problems implies first to control combustion technologies since they are the major source of the problem and the first place to act. Considering that there is no real substitute for combustion at the moment in many applications (aircrafts, cars, energy production), it also means that the optimization of combustion processes is the most effective method to control global change.

The optimization of combustion is an ongoing work since 1900 but recent progress in this field has been tremendous. In the last twenty years, combustion devices have been optimized in terms of efficiency and pollution emissions to reach norms which were impossible to imagine before. This has been done by the introduction of electronic monitoring and control (especially for car engines) but also by a better understanding of combustion phenomena and an optimization of the parameters of combustion chambers. These parameters are not limited to the combustion chamber shape: the fuel injection strategy, for example, is a key point to control combustion. Optimizing a combustion chamber is therefore an extremely difficult process and this complexity is obvious when one considers the results of these optimization processes in combustion companies: while the shapes of most civil aircrafts today look the same, all combustion chambers are different showing that the optimum is by no means simple to define.

What makes combustor optimization even more difficult is the multiple non-linearities and instabilities found in reacting flows:

– Minimizing pollutant is easy to obtain by simply injecting less fuel in a chamber. The problem however is that, below a certain equivalence ratio (below a certain amount of kg of fuel per kg of air), combustion simply stops [3, 7, 4]. The existence of this flammability limit makes optimization delicate because bringing the combustor close to extinction is

dangerous (for aircrafts and helicopters for example, this is definitely something which must be avoided for obvious reasons).
- Optimization of combustion devices must be sought for a whole range of operating conditions. Many chambers can be optimized at one regime (for example idle conditions in a car) but then will not be efficient for another regime (full power for example). Moreover, a chamber can be optimized for a regime (a gas turbine for example) but impossible to ignite or too sensitive to sudden quenching.
- The most critical problem encountered since the end of the 20th century in the field of gas turbines is combustion instabilities [8–10, 4]. Most chambers which were optimized to minimize NOx emissions and maximize efficiency in the last ten years have been subject to combustion instability problems. In Europe, the LOW NOx projects initiated by the European Commission are now being continued through combustion instability studies because the gains in NOx and efficiency are often compromised by the impact of combustion instabilities. Section 3 will focus on this specific issue.

3 Combustion and instabilities

Reacting flows are compressible flows. They exhibit acoustic / combustion instabilities which can be extremely strong [3, 11, 4]. The fact that flames can couple with acoustics has been known for a long time [12] even though it is still not fully understood. Combustion instabilities are difficult to predict and are usually discovered at a late stage during the development of engine programmes so that they represent a significant industrial risk.

In steady combustors like gas turbines, instabilities can lead to oscillations of all flow parameters, reaching levels which are incompatible with the normal operation of the chamber. High levels of structure oscillations are found, very high levels of RMS pressure can be observed. In a given chamber, while normal turbulent combustion usually leads to 10 to 100 Pa RMS pressure levels, it is not uncommon to see chambers where the RMS pressure reaches 20000 Pa (180 dB) when a combustion instability begins. At these levels, the acoustic velocity associated to the RMS pressure can reach 1 to 20 m/s so that the perturbations induced by the acoustic field are absolutely not negligible. In such cases, the engine structure can fail, the fuel injector can burn, the flame might totally quench or flashback. Flashback is a phenomenon encountered when the acoustic velocity is larger than the mean flow leading to flow reversal at the combustor inlet: in other words, the flow leaves the combustor through the inlet instead of entering it; the flame does the same and ends up upstream of the combustion chamber, in a zone which was not designed to sustain high temperatures. Combustion instabilities have been the source of multiple failures in rocket engines, as early as the Saturne or the Ariane 4 project, in aircraft engines (main chamber of post combustion chamber), in industrial gas turbines, in industrial furnaces, etc. Fig. 1 shows an example of simulation of 'mild' oscillation in a gas turbine[13] where the flame position (visualized by an isosurface of temperature colored by axial velocity) pulsates strongly at four instants of a cycle occuring at 120 Hz). For such a mild oscillation, a limit cycle is obtained and the chamber can operate for a long time without problem except for a high noise level.

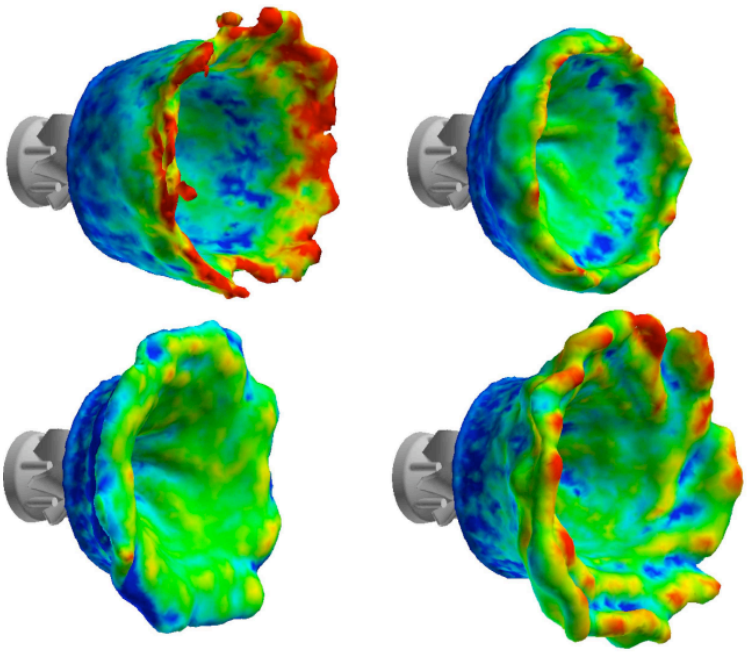


Fig. 1: Snapshots of flame position (isosurface of temperature) during one oscillation cycle at 120 Hz in an industrial gas turbine [13]. LES result.

Predicting and controlling combustion instabilities is a major challenge for combustion research. Today, the most promising path is to understand these phenomena using Large Eddy Simulation methods which are able to predict these combustion oscillations [4, 14, 15] something which was impossible 10 years ago with classical Reynolds Averaged methods.

4 DNS, LES and RANS for combustion

Turbulent combustion is encountered in most practical combustion systems such as rockets, internal combustion or aircraft engines, industrial burners and furnaces... while laminar combustion applications are almost limited to candles, lighters and some domestic furnaces. Studying and modeling turbulent combustion processes is therefore an important issue to develop and improve practical systems (*i.e.* to increase efficiency and reduce fuel consumption and pollutant formation). As combustion processes are difficult to handle using analytical techniques, numerical combustion for turbulent flames is a fast growing area.

The three main numerical approaches used in turbulent combustion modeling are Reynolds Averaged Navier Stokes (RANS) where all turbulent scales are modelled, Direct Numerical

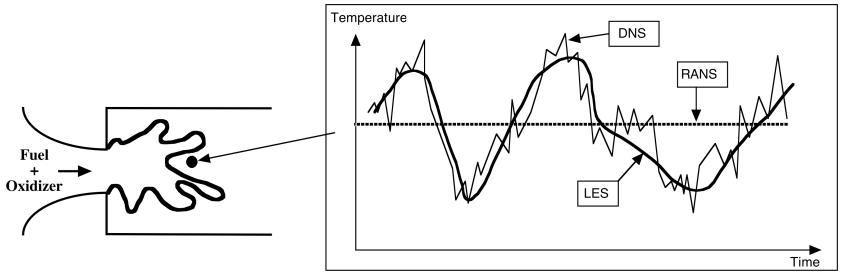


Fig. 2: Examples of time evolutions of the local temperature computed with DNS, RANS or LES in a turbulent flame brush.

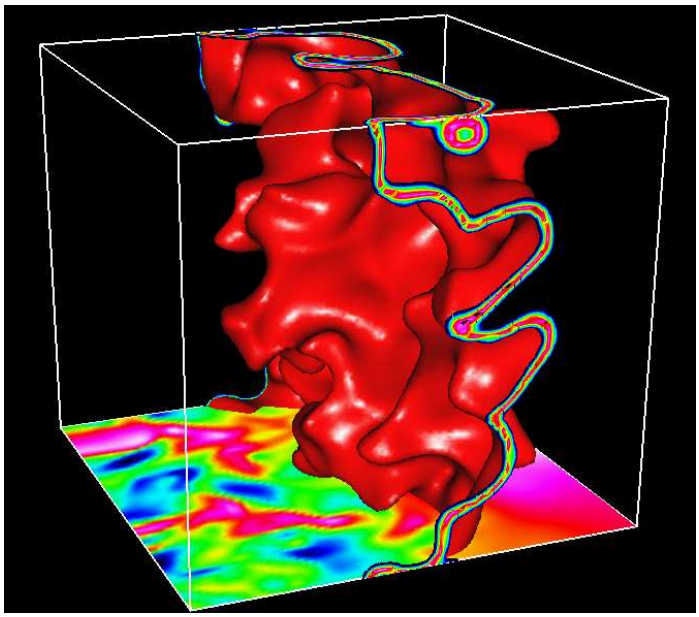


Fig. 3: DNS of a premixed flame interacting with three-dimensional isotropic turbulence [16]. An isosurface of temperature is visualized. The reaction rate is presented in two planes which are normal to the mean flame front. The vorticity field, corresponding to turbulent motions, is also displayed in the bottom plane.

Simulation (DNS) where all scales are resolved and Large Eddy Simulation (LES) where larger scales are explicitly computed whereas the effects of smaller ones are modeled:

- Reynolds Averaged Navier Stokes (or RANS) computations have historically been the first possible approach because the computation of the instantaneous flow field in a turbulent flame was impossible. Therefore, RANS techniques were developed to solve for the mean values of all quantities. The balance equations for Reynolds or Favre (*i.e.* mass-weighted) averaged quantities are obtained by averaging the instantaneous balance equations. The averaged equations require closure rules: a turbulence model to deal with the flow dynamics in combination with a turbulent combustion model to describe chemical species conversion and heat release. Solving these equations provides averaged quantities corresponding to averages over time for stationary mean flows or averages over different realizations (or cycles) for periodic flows like those found in piston engines (*i.e.* phase averaging). For a stabilized flame, the temperature predicted with RANS at a given point is a constant corresponding to the mean temperature at this point (Fig. 2).
- The second level corresponds to Large-Eddy simulation (LES). The large vortices are explicitly calculated whereas the smaller ones are modeled using subgrid closure rules. The balance equations for large-eddy simulations are obtained by filtering the instantaneous balance equations. LES determine the instantaneous position of a "large scale" resolved flame front but a subgrid model is still required to take into account the effects of small turbulent scales on combustion. LES would capture the low-frequency variations of temperature (Fig. 2).
- The third level of combustion simulations is Direct Numerical Simulation (DNS) where the full instantaneous Navier-Stokes equations are solved without any model for turbulent motions: all turbulence scales are explicitly determined and their effects on combustion are captured. DNS would predict all time variations of temperature (Fig. 2) exactly like a high-resolution sensor would measure them in an experiment (Fig. 3). Developed in the last twenty years thanks to the development of high performance computers, DNS have changed the analysis of turbulent combustion but are still limited to simple academic flows (*i.e.* simple geometries and somewhat low Reynolds numbers).

In terms of computational requirements, CFD for non-reacting and reacting flows follow similar trends: DNS is the most demanding method and is limited to fairly low Reynolds numbers and simplified geometries. LES works with coarser grids (only larger scales have to be resolved) and may be used to deal with higher Reynolds numbers but require subgrid-scale models. In current engineering practice, RANS is extensively used because it is less demanding in terms of resources.

5 Massively parallel LES of combustion

Performing LES of real devices requires extremely large parallel machines. The solver used in this paper is an explicit solver called AVBP and developed by CERFACS and Institut Francais du Pétrole [17–19] for multiple industrial users and research laboratories (www.cerfacs.fr/cfd/parallel.php).

AVBP solves the compressible Navier Stokes equations in a multispecies gas with chemical reactions [4]. This implies typically advancing 10 variables (3 velocities, density, temperature and five chemical species) on a 10 million grid points over 1 million time iterations. The usual duration of a computation in physical time is of the order of 0.5 seconds for a real combustion chamber. This time is sufficient to understand instabilities which can develop in these devices. In terms of human time, such a computation can take from one day to one month.

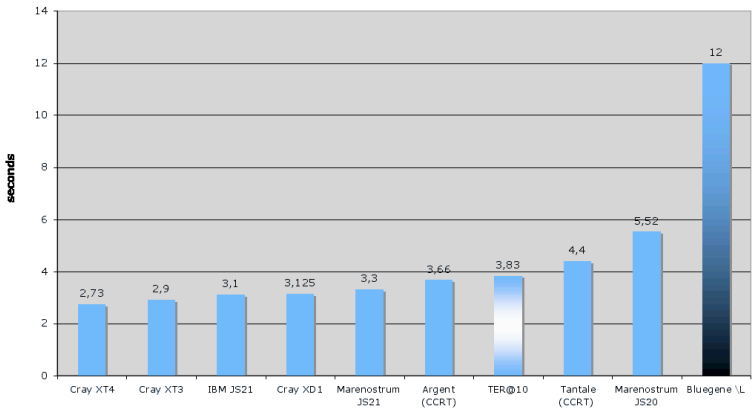


Fig. 4: Elapsed CPU times for AVBP to perform one time iteration on a 10 million point grid.

Over the last ten years, AVBP has been applied succesfully to multiple non reacting flows [20–22], piston engine configurations [23, 24], academic combustors [25–28], combustion instabilities [29, 30, 15], ramjet engines [31] and real gas turbines [32, 33]. A key aspect of the success of AVBP is its capability to make use of all existing parallel machines with very limited adaptations.

To achieve efficiency on parallel architectures, AVBP uses MPI for message passing and HDF 5 for data format. Since the beginning of the AVBP project, two main choices have been made in order to be ready for massively parallel machines:

- LES solvers are unsteady solvers: they advance the solution in time. This is done with an explicit method to be able to scale on very large number of processors.
- The data structure is built for hybrid meshes (structured and unstructured) to allow easy mesh decomposition even on large number of processors.

As a result, AVBP has been used on multiple parallel architectures: Fig. 4 shows typical CPU times required to perform one time iteration on a 10 million cell grid using 32 processors. Of course, machines like BlueGene provide less efficient results when the number of processors is limited (here to 32). However, our objective is not to run on 32 processors but on 4000

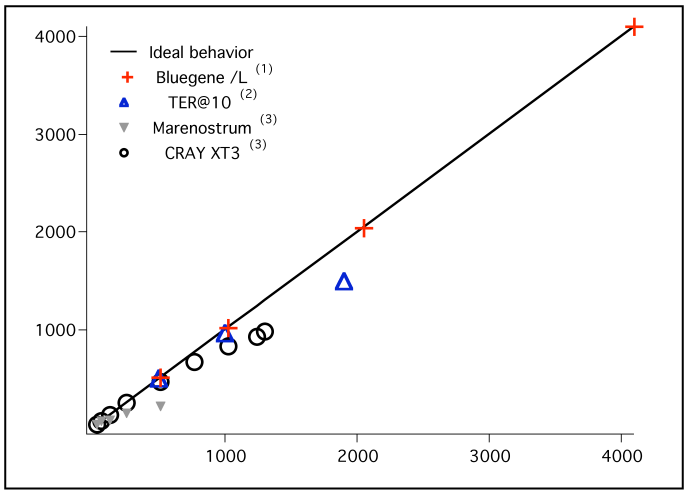


Fig. 5: Speed ups of AVBP measured on various parallel architectures.

or 40000. In this case, Fig. 5 shows that the parallel efficiency of machines like BlueGene is very high and that AVBP scales almost perfectly. This is a key issue when performing runs on machines with more than 100,000 processors (see for example the INCITE program at www.sc.doe.gov/ascr/incite/index.html)

6 Rounding errors and LES

The literature shows the power of Large Eddy Simulation (LES) to predict non-reacting [34, 35] as well as reacting turbulent flows[36, 4, 37–39, 28]. The previous sections have suggested that the main strength of LES compared to classical Reynolds Averaged (RANS) methods is that, like Direct Numerical Simulation (DNS)[40–42], LES explicitely captures large scale unsteady motions due to turbulence instead of modeling them. An often ignored aspect of this feature is that like DNS, LES is also submitted to a well-known feature of turbulent flows: the exponential separation of trajectories [43] implies that the flow solution exhibited by LES is very sensitive to any "small perturbation" with respect to a reference state. These small perturbations which can induce new 'instabilities' can have different sources. Rounding errors are the first source of random noise in any finite precision computation: they constitute an unavoidable forcing for the Navier-Stokes equations and may lead to LES variability. The study of error growth in finite precision computations is an important topic in applied mathematics [44, 45] but has found few applications in multidimensional fluid mechanics because of the complexity of the codes used in CFD.

Initial conditions are a second source of LES result variabilities: these conditions are often unknown and any small change in initial conditions may trigger significant changes in the LES solution.

Due to its large computational resource requirements, modern LES heavily relies on parallel computing. However, in codes using domain decomposition, *i.e.* most of them, it is also an additional "noise" source in the Navier-Stokes equations especially at partition interfaces. Even in explicit codes where the algorithm is independent of the number of processors, the different summation orders with which a nodal value is reconstructed at partition interfaces may induce non-associativity errors. For example, in explicit codes on unstructured meshes using cell vertex methods[17], the residual at one node is obtained by adding the weighted residuals of the surrounding cells. Additions of only two summands are perfectly associative. Moreover, it must be noted that not all additions of more than two summands generate non-associativity errors. However, in some cases summation may yield distinct results for floating-point accumulation: the rounding errors in $(a+b)+c$ and in $a+(b+c)$ may be different, in particular if there are large differences in orders of magnitude between the terms[46]. After thousands of iterations, the LES result may be affected. Since these rounding errors are induced by non deterministic message arrival at partition interfaces, it is believed that such behaviour may occur for any unstructured parallel CFD code, regardless of the numerical schemes used. As a consequence, the simulation output might change when run on a different number of processors. The case of implicit codes in time[36, 35, 47] or in space such as compact schemes[48–50] is not considered here: for such schemes, the methods [51, 52] used to solve the linear system appearing at each iteration depend on the number of processors. Therefore, rounding errors are not the only reason why solutions obtained with different numbers of processors differ. Even on a single processor computation, internal parameters of the partitioning algorithm may couple with rounding errors to force the LES solution. For example, a different reordering of nodes using the Cuthill-McKee (CM) or the reverse Cuthill-McKee (RCM) algorithm[53, 54] may produce the same effect as a simple perturbation and can be the source of solution divergence.

Turbulence theory indicates that LES/DNS solutions have a meaning only in a statistical manner [55]: observing that the solution of a given LES/DNS at a given instant changes when the rounding errors or the initial conditions change is not really surprising. It becomes a real difficulty in the practical use of LES/DNS because running the same simulation on two different machines or one machine with a different number of processors or slightly different initial conditions can lead to totally different instantaneous results. For steady flows in the mean, statistics do not depend on these changes and mean profiles will be identical. However, when the objective of the LES is the study of unsteady phenomena such as ignition or quenching in a combustor[26], knowing that results depend on these parameters is certainly a sobering thought for the LES/DNS community and a drawback in terms of industrial exploitation.

This last section addresses these issues and tries to answer a simple question which is of interest for all practitioneers of LES: how does the solution produced by LES depend on the number of processors used to run the simulation? On the initial condition? On internal details of the algorithm?

First, we will present an example of the effects of the number of processors in a simple case: a rectangular turbulent channel computed with a fully explicit LES code[19]. This example shows that even in an explicit code, running a simulation twice on a different number of processors can lead to totally different instantaneous solutions. The second section then gives a systematic description of the effects of rounding errors in two flows: a turbulent channel and a laminar Poiseuille flow. For all cases, the difference between two instantaneous solutions obtained by changing either the number of processors, the initial condition or the graph ordering is quantified in terms of norms between the two solutions. The effects of time step and machine precision (simple, double and quadruple) are also investigated.

6.1 Effects of the number of processors on LES

This first example is the LES of a rectangular fully developed turbulent channel of dimensions: 75x25x50 mm (Fig. 6). A pressure gradient is applied to a periodic channel flow and random disturbances are added to pass transition to turbulence. There are no boundary conditions except for the walls. The Reynolds number is $Re_\tau = \delta u_\tau/\nu = 1500$, where δ is half the channel height and u_τ the friction velocity at the wall: $u_\tau = (\tau_{wall}/\rho)^{1/2}$ with τ_{wall} being the wall stress. The mesh contains 30^3 hexahedral elements, it is not refined at walls. The first grid point is at a reduced distance $y^+ = yu_\tau/\nu \approx 100$ of the wall. The subgrid model is the Smagorinsky model and a law-of-the-wall is used at the walls[28]. The CFL number λ controlling the time step Δt is $\lambda = max((u+c)\Delta t/\Delta)$ where u is the local convective velocity, c the speed of sound and Δ the mesh size. For all simulations discussed below, the initial condition corresponds to a snapshot of the flow at a given instant, long after turbulence was initialized so that it is fully established. The computation is performed with an explicit code where domain decomposition is such that the method is perfectly equivalent on any number of processors. The Recursive Inertial Bisection (RIB)[56, 57] algorithm has been used to partition the grid and the Cuthill-McKee algorithm is considered as the default graph reordering strategy. The scheme used here is the Lax-Wendroff scheme[58]. Additional tests were performed using a third-order scheme in space and time[18] but led to the same conclusions.

Figs. 7–9 show fields of axial velocity in the central plane of the channel at three instants after the run initialization. Two simulations performed on respectively 4 (TC1) and 8 processors (TC2) with identical initial conditions are compared. The characterictics of all presented simulations are displayed in Table 1 and 2. The instants correspond to (in wall units) $t^+ = 7.68$, $t^+ = 18.43$ and $t^+ = 26.11$ respectively where $t^+ = u_\tau t/\delta$. Obviously, the two flow fields observed at $t^+ = 7.68$ are identical. However, at $t^+ = 18.43$, differences start to become visible. Finally, at $t^+ = 26.11$, the instantaneous flow fields obtained in TC1 and TC2 are totally different. Even though the instantaneous flow fields are different, statistics remain the same: mean and root mean square axial velocity profiles averaged over $t^+ \approx 60$ are identical for both simulations.

This very simple example illustrates the main question of the present work: is the result of Figs. 7–9 reasonable? If it is not a simple programming error (the next section will show that it is not), can other parameters produce similar effects?

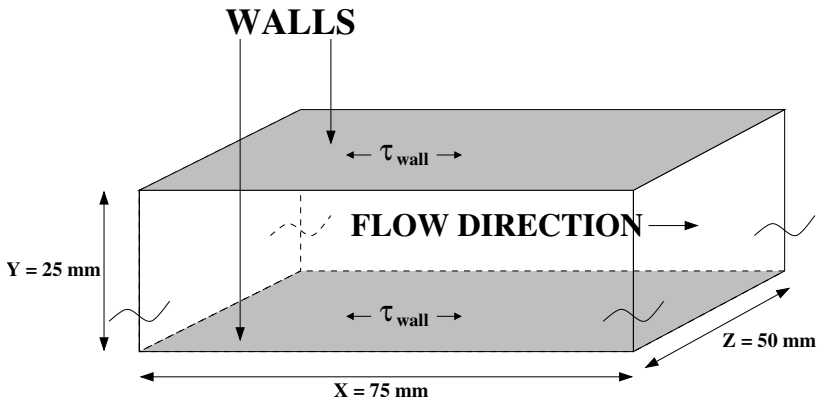
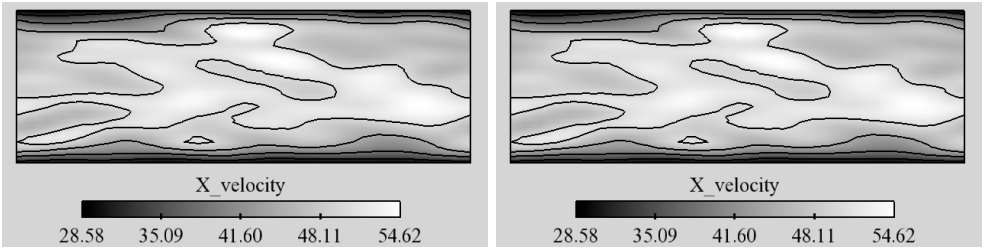
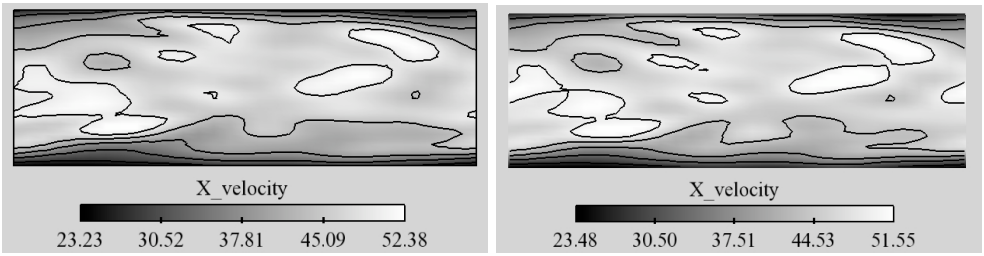


Fig. 6: Schematic of a periodic channel. The upper and lower boundaries consist of walls, all other boundaries are pairwise periodic.



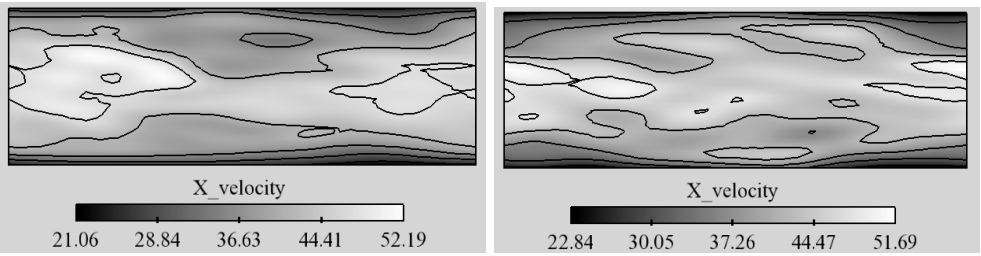
(a) Run with 4 processors (b) Run with 8 processors

Fig. 7: Instantaneous field of axial velocity in the central plane of the channel at $t+ = 7.68$. a) run TC1 (4 processors), b) run TC2 (8 processors).



(a) Run with 4 processors (b) Run with 8 processors

Fig. 8: Instantaneous field of axial velocity in the central plane of the channel at $t+ = 18.43$. a) run TC1 (4 processors), b) run TC2 (8 processors).



(a) Run with 4 processors (b) Run with 8 processors

Fig. 9: Instantaneous field of axial velocity in the central plane of the channel at $t+ = 26.11$. a) run TC1 (4 processors), b) run TC2 (8 processors).

6.2 Sensitivity of LES in laminar and turbulent flows

To understand how LES can produce diverging instantaneous results such as those shown in the previous section, simple tests were performed to investigate the effects of various aspects of the methodology:

- laminar/turbulent baseline flow,
- number of processors,
- initial condition,
- graph ordering,
- time step,
- machine precision.

For these tests, the objective is to quantify the differences between two LES solutions produced by a couple of simulations in Table 1 and 2. Let u_1 and u_2 be the scalar fields of two given instantaneous solutions at the same instant after initialization. A proper method to compare the latter is to use the following norms:

$$N_{max} = \max(u_1(\mathbf{x}) - u_2(\mathbf{x})) \text{ for } x \in \Omega N_{mean} = (\frac{1}{V_\Omega} \int_\Omega (u_1(\mathbf{x}) - u_2(\mathbf{x}))^2 d\Omega)^{\frac{1}{2}} \text{ for } x \in \Omega \quad (1)$$

where Ω and V_Ω respectively denote the computational domain and its volume. Both norms (in m/s) will be applied to the axial velocity field so that N_{max} provides the maximum local velocity difference in the field between two solutions while N_{mean} yields a volumetrically averaged difference between the two solutions. The growth of N_{max} and N_{mean} versus the number of iterations will be used as a direct indicator for the divergence of the solutions.

6.3 A fully deterministic LES?

First, it is useful to indicate that performing any of the LES of Table 1 twice on the same machine with the same number of processors, the same initial conditions and the same partition algorithm leads to exactly the same solution, N_{max} and N_{mean} being zero to machine

Table 1: Summary of turbulent LES runs (fully developed turbulent channel).

Run Id	Nbr proc	Init. cond.	Precision	Graph ordering	CFL λ
TC1	4	Fixed	Double	CM	0.7
TC2	8	Fixed	Double	CM	0.7
TC3	1	Fixed	Double	CM	0.7
TC4	1	Modif.	Double	CM	0.7
TC5	1	Fixed	Double	RCM	0.7
TC6	4	Fixed	Double	CM	0.35
TC7	8	Fixed	Double	CM	0.35
TC8	4	Fixed	Simple	CM	0.7
TC9	8	Fixed	Simple	CM	0.7
TC10	28	Fixed	Quadr.	CM	0.7
TC11	32	Fixed	Quadr.	CM	0.7

accuracy. In that sense, the LES remains fully deterministic. However, this is true only if the order of operations at interfaces is not determined by the order of message arrival so that summations are always carried out in the same order. Otherwise, the randomness induced by the non deterministic order of message arrival is enough to induce diverging solutions. Note that such an option can be expensive and that blocking messages order can increase the overall simulation cost by a large amount.

6.4 Influence of turbulence

The first test is to compare a turbulent channel flow studied in the previous section and a laminar flow. A three dimensional Poiseuille flow was used as test case. The Poiseuille computation is performed on a pipe geometry with 361 by 26 points. The flow is laminar and the Reynolds number based on the bulk velocity and diameter is approximately 500. The boundary conditions are set periodic at the inlet/outlet and no slip at the duct walls, a constant axial pressure gradient is imposed in the entire domain.

Table 2: Summary of laminar runs (Poiseuille flow).

Run Id	Nbr proc	Init. cond.	Precision	Graph ordering	CFL λ
LP1	4	Fixed	Double	CM	0.7
LP2	8	Fixed	Double	CM	0.7

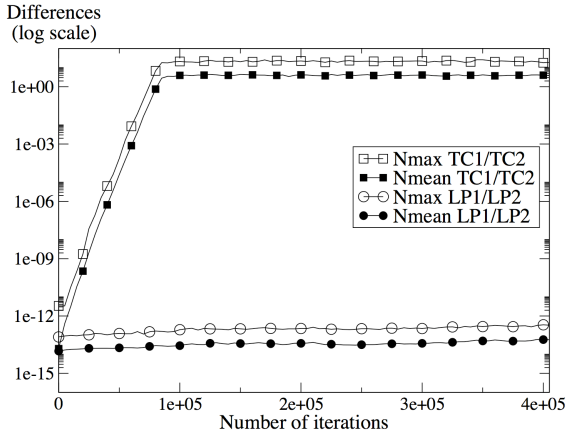


Fig. 10: Effects of turbulence. Differences between solutions measured by N_{max} (open symbols) and N_{mean} (closed symbols) versus iteration. Squares: differences between TC1 and TC2 (turbulent channel). Circles: differences between LP1 and LP2 (laminar Poiseuille flow).

Figure 10 shows the evolutions of N_{max} and N_{mean} versus iteration for runs TC1/TC2 and LP1/LP2. Note that the first point of the graph is the evaluation of the difference after one iteration. The only parameter tested here is a change of the number of processors. As expected from the snapshots of Figs. 7–9, the turbulent channel simulations are very sensitive to a change in the number of processors and the solutions of TC1 and TC2 diverge rapidly leading to a maximum difference of 20 m/s and a mean difference of 3-4 m/s after 90,000 iterations. On the other hand, the difference between LP1 and LP2 hardly increases and levels off when reaching values of the order or 10^{-12}. This is expected since there is obviously only one stable solution for the Poiseuille flow for infinite times and laminar flows do not induce exponential divergence of trajectories. However, this simple test case confirms that the turbulent character of the flow is the source of the divergence of solutions. This phenomenon must not be confused with the growth of a hydrodynamic mode, which is induced by the bifurcation in phase space of an equilibrium state of a given physical system. Obviously, such an equilibrium state does not exist for a fully developed turbulent channel flow. In this case, the separation of trajectories is caused by vorticity, which leads to an increase in the number of degrees of freedom in phase space [59] and thus high sensitivity to initial conditions. Moreover, the stagnation of absolute and mean differences between TC1/TC2 simply implies that after 90,000 iterations solutions have become fully uncorrelated and should not be misinterpreted as the saturation of an exponentially growing mode.

The basic mechanism leading to Figs. 7–9 is that the turbulent flow acts as an amplifier for rounding errors generated by the fact that the mesh is decomposed differently in TC1 and TC2. The source of this difference is the new graph reordering obtained for both decompositions. This implies a different ordering when adding the contributions to a cell residual for nodes

inside the subdomains but mainly at partition interfaces. This random noise roughly starts at machine accuracy (Fig. 10) at a few points in the flow and grows continuously if the flow is turbulent.

6.5 Influence of initial conditions

The previous section has shown that turbulence combined with a different domain decomposition (i.e. a different number of processors for the following) is sufficient to lead to totally different instantaneous flow realizations. It is expected that a perturbation in initial conditions will have the same effect as domain decomposition. This is verified in runs TC3 and TC4 which are run on one processor only, thereby eliminating issues linked to parallel implementation. The only difference between TC3 and TC4 is that in TC4, the initial solution is identical to TC3 except at one random point where a 10^{-16} perturbation is applied to the streamwise velocity component. Simulations with different locations of the perturbation were run to ensure that their position did not affect results.

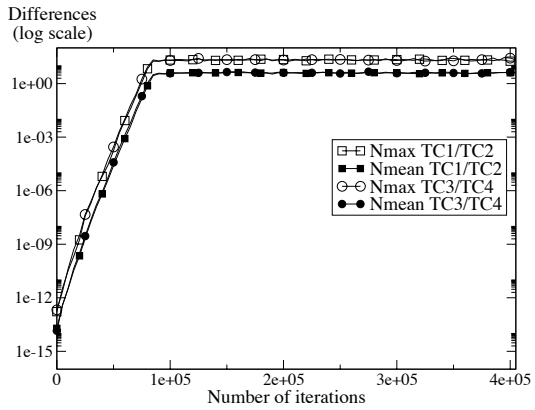


Fig. 11: Effects of initial conditions. Differences between solutions measured by N_{max} (open symbols) and N_{mean} (closed symbols) versus iteration. Squares: differences between TC1 and TC2 (different numbers of processors). Circles: differences between TC3 and TC4 (different initial conditions).

Figure 11 shows that the growth rate of the difference between TC3 and TC4 is exactly the same as the one observed between TC1 and TC2 (also displayed in Fig. 11): two solutions starting from a very slightly perturbed initial condition diverge as fast as two solutions starting from the same solution but running on different numbers of processors. Note that the difference between runs TC1 and TC2 comes from random rounding errors introduced at each time step

while TC3 and TC4 differ only through the initial condition: no perturbation is added during the simulation. Still, the differences between TC3 and TC4 increase as fast as those between TC1 and TC2: this confirms that a turbulent flow amplifies any difference in the same manner, whether it is due to rounding errors or to a perturbation of the initial conditions.

6.6 Effects of graph ordering

It has already been indicated that performing the same simulation twice (with the same number of processors and same initial conditions) leads to exactly the same result. However, this is only true as long as exactly the same code is used. It is not verified any more as soon as a modification affecting rounding errors is done in the code. At this point, so many factors affecting rounding errors can be cited that a general discussion is pointless. This paper will focus on fully explicit codes and on one example only: the order used to add residuals at nodes in a cell vertex scheme. This order is controlled by the developer. For simulation TC5, the ordering of this addition was changed (reverse Cuthill-McKee algorithm): the residual at a given mesh node was assembled by adding the contributions to a cell residual in a different order. This change does not affect the flow data: in TC5 the node residual in a regular tetrahedral mesh is obtained by $1/4(R_1 + (R_2 + (R_3 + R_4))$ where the R_i's are the residuals of the cells surrounding the node and by $1/4(R_4 + (R_3 + (R_2 + R_1))$ in TC3. It has an effect, however, on rounding errors and the cumulated effects of this non-associativity error are what this test tries to isolate. TC5 and TC3 are performed with the same initial condition and run on one processor only. The only difference is the graph reordering strategy.

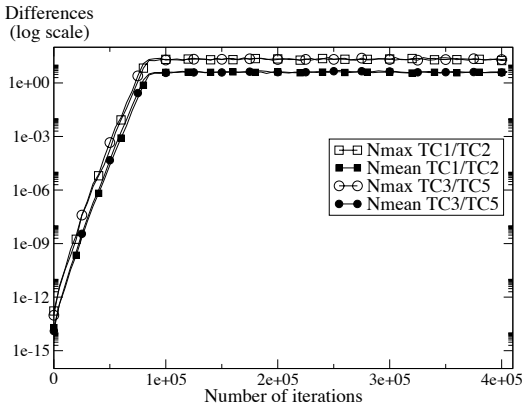


Fig. 12: Effects of addition order. Differences between solutions measured by N_{max} (open symbols) and N_{mean} (closed symbols) versus iteration. Squares: differences between TC1 and TC2. Circles: differences between TC3 and TC5.

As shown by Fig. 12, the differences between TC5 and TC3 are again similar to those observed between TC1 and TC2 (obtained by changing the number of processors). This confirms that rounding errors (and not the parallel character of the code) are the source of the solution divergence. It also shows that any modification of the code could lead to such a divergence, suggesting that repeating an LES simulation with the same code after a few months and a few modifications will probably never yield the same instantaneous flow fields, potentially leading to discussions on the validity of the modified code.

6.7 Effects of time step

It is interesting to verify that numerical aspects do not influence the growth rate of the solutions difference and that the growth rate is only determined by the physical and geometrical parameters of the configuration. On that account, simulations TC6 and TC7 are performed with a time step reduced by a factor 2 compared to simulations TC1 and TC2. TC6 and TC7 are carried out on respectively 4 and 8 processors. The norms between TC6 and TC7 are displayed in Fig. 13 and compared to the norms between TC1 and TC2. From the explanations given above, similar growth rates are expected when comparing the growth rates over physical time. The growth rates observed in Fig. 13 are indeed very similar. The slight difference is probably due to the variation of the numerical dispersion and dissipation properties of the scheme with the time step [58].

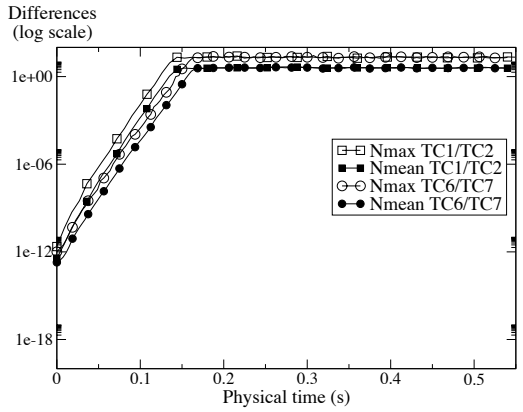


Fig. 13: Effects of time step. Differences between solutions measured by N_{max} (open symbols) and N_{mean} (closed symbols) versus physical time. Squares: differences between TC1 and TC2 (time step Δt). Circles: differences between TC6 and TC7 (time step $\Delta t/2$).

6.8 Effects of machine precision

A last test to verify that the divergence between solutions is not due to a programming error but depends primarily on rounding errors is to perform the same computation with simple/quadruple precision instead of double precision. Simulations TC1 and TC2 were repeated using simple precision in runs TC8 and TC9 (Table 1) and quadruple precision in TC10 and TC11. To compensate for the increase in computational time for quadruple precision simulations, roughly a factor ten compared to double precision, TC10 and TC11 were carried out on respectively 28 and 32 processors in order to yield a reasonable restitution time. Results are displayed in Fig. 14 and compared to the difference between TC1 and TC2.

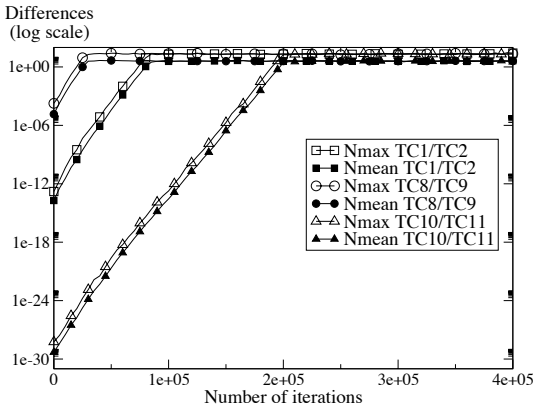


Fig. 14: Effects of machine accuracy. Differences between solutions measured by N_{max} (open symbols) and N_{mean} (closed symbols) versus iteration. Squares: differences between TC1 and TC2 (double precision). Circles: differences between TC8 and TC9 (simple precision). Triangles: differences between TC10 and TC11 (quadruple precision)

Figure 14 shows that the solutions differences for TC8/TC9 and TC10/TC11 roughly start from the respective machine accuracies (differences of 10^{-6} for simple precision after one iteration, differences of 10^{-30} for quadruple precision after one iteration) and increase exponentially with the same growth rate before reaching the same difference levels for all three cases. This shows that higher precision computations cannot prevent the exponentional divergence of trajectories but only delay it.

7 Conclusions

This paper has shown the power of Large Eddy Simulation to understand combustion instabilities and has focused on the efficiency of modern parallel solvers for reacting flows. A new specific question raised by these solvers for turbulent flows is the sensitivity of instantaneous LES fields to multiple parameters such as number of processors, initial condition, time step, changes in addition ordering of cell residuals for cell vertex methods. The baseline simulation used for the tests was a fully developed turbulent channel. Results confirm that any turbulent flow computed in LES exhibits significant sensitivity to these parameters, leading to instantaneous solutions which can be totally different. Laminar flows are almost insensitive to these parameters. The divergence of solutions is due to two combined facts: (1) the exponential separation of trajectories in turbulent flows and (2) the non-deterministic rounding errors induced by different domain decompositions or different ordering of operations. More generally any change in the code lines affecting rounding errors will have the same effects. Similarly, small changes in initial condition (of the order of machine accuracy at one point of the flow only) produce a divergence of solutions. Working with higher precision machines does not suppress the divergence of solutions but delays it.

These results confirm the expected nature of LES [55] in which solutions are meaningful only in a statistical sense and instantaneous values can not be used for analysis. However, on a more practical level, they point out various difficulties to develop LES codes: (1) repeating the results of a given LES after modifying the code and verifying that instantaneous solutions have not changed is not always possible. Since any programming error will also lead to a change in instantaneous solutions, identifying errors introduced by new lines will require a detailed analysis based on average fields (and not on instantaneous fields) and a significant loss of time. (2) Verifying a LES code on a parallel machine is a difficult task: running the code on different numbers of processors will lead to different solutions and make comparisons impossible. (3) Porting a LES code from one machine to another will also produce different solutions for turbulent runs, making comparison and validations of new architectures difficult.

The concept of "quality" in LES requires obviously more detailed studies and tools than what has been used up to now in Reynolds Averaged simulations. Instabilities appearing in a given LES on a given computer can have sources which were not expected at first sight (like the number of processors). Mastering these instabilities (or at least understanding them) will be an important task to get the full power of LES techniques.

References

1. Drazin, P.G., Reid, W.H.: Hydrodynamic stability. Cambridge University Press, London (1981)
2. Ho, C.M., Huerre, P.: Perturbed free shear layers. J. Fluid Mech.**16** (1984) 365
3. Williams, F.A.: Combustion theory. Benjamin Cummings, Menlo Park, CA (1985)
4. Poinsot, T., Veynante, D.: Theoretical and numerical combustion. R.T. Edwards, 2nd edition. (2005)
5. Vichnevetsky, R., Bowles, J.B.: Fourier analysis of numerical approximations of hyperbolic equations. SIAM Studies in Applied Mechanics, Philadelphia (1982)

6. Sengupta, T.K.: Fundamentals of Computational Fluid Dynamics. Universities Press, Hyderabad (India) (2004)
7. Kuo, K.K.: Principles of Combustion. John Wiley, New York (1986)
8. Candel, S.: Combustion instabilities coupled by pressure waves and their active control. In: 24th Symp. (Int.) on Combustion, The Combustion Institute, Pittsburgh (1992) 1277–1296
9. McManus, K., Poinsot, T., Candel, S.: A review of active control of combustion instabilities. Prog. Energy Comb. Sci.**19** (1993) 1–29
10. Lieuwen, T., Yang, V.: Combustion instabilities in gas turbine engines. operational experience, fundamental mechanisms and modeling. In: Prog. in Astronautics and AeronauticsAIAA. Volume 210. (2005)
11. Crighton, D.G., Dowling, A.P., Williams, J.E.F., Heckl, M., Leppington, F.: Modern methods in analytical acoustics. Lecture Notes. Springer Verlag, New-York (1992)
12. Rayleigh, L.: The explanation of certain acoustic phenomena. Nature **July 18** (1878) 319–321
13. Giauque, A., Selle, L., Poinsot, T., Buechner, H., Kaufmann, P., Krebs, W.: System identification of a large-scale swirled partially premixed combustor using LES and measurements. J. Turb.**6**(21) (2005) 1–20
14. Martin, C., Benoit, L., Sommerer, Y., Nicoud, F., Poinsot, T.: Les and acoustic analysis of combustion instability in a staged turbulent swirled combustor. AIAA Journal**44**(4) (2006) 741–750
15. Selle, L., Benoit, L., Poinsot, T., Nicoud, F., Krebs, W.: Joint use of compressible large-eddy simulation and helmoltz solvers for the analysis of rotating modes in an industrial swirled burner. Combust. Flame**145**(1-2) (2006) 194–205
16. Boger, M., Veynante, D., Boughanem, H., Trouvé, A.: Direct numerical simulation analysis of flame surface density concept for large eddy simulation of turbulent premixed combustion. In: 27th Symp. (Int.) on Combustion, Boulder, The Combustion Institute, Pittsburgh (1998) 917–927
17. Schönfeld, T., Rudgyard, M.: Steady and unsteady flows simulations using the hybrid flow solver avbp. AIAA Journal**37**(11) (1999) 1378–1385
18. Colin, O., Rudgyard, M.: Development of high-order taylor-galerkin schemes for unsteady calculations. J. Comput. Phys.**162**(2) (2000) 338–371
19. Moureau, V., Lartigue, G., Sommerer, Y., Angelberger, C., Colin, O., Poinsot, T.: High-order methods for DNS and LES of compressible multi-component reacting flows on fixed and moving grids. J. Comput. Phys.**202**(2) (2005) 710–736
20. Prière, C., Gicquel, L.Y.M., Kaufmann, A., Krebs, W., Poinsot, T.: Les of mixing enhancement: LES predictions of mixing enhancement for jets in cross-flows. J. Turb.**5** (2004) 1–30
21. Prière, C., Gicquel, L.Y.M., Gajan, P., Strzelecki, A., Poinsot, T., Bérat, C.: Experimental and numerical studies of dilution systems for low emission combustors. Am. Inst. Aeronaut. Astronaut. J.**43**(8) (2005) 1753–1766
22. Mendez, S., Nicoud, F.: Large-eddy simulation of a bi-periodic turbulent flow with effusion. J. Fluid Mech., in press (2008)
23. Moureau, V.R., Vasilyev, O.V., Angelberger, C., Poinsot, T.J.: Commutation errors in Large Eddy Simulations on moving grids: Application to piston engine flows. In: Proc. of the Summer Program, Center for Turbulence Research, NASA AMES/Stanford University, USA (2004) 157–168
24. Richard, S., Colin, O., Vermorel, O., Benkenida, A., Angelberger, C., Veynante, D.: Towards large eddy simulation of combustion in spark ignition engines. Proc. Combust. Inst.**31**(3059-3066) (2007)

25. Selle, L., Lartigue, G., Poinsot, T., Koch, R., Schildmacher, K.U., Krebs, W., Prade, B., Kaufmann, P., Veynante, D.: Compressible large-eddy simulation of turbulent combustion in complex geometry on unstructured meshes. Combust. Flame**137**(4) (2004) 489–505
26. Sommerer, Y., Galley, D., Poinsot, T., Ducruix, S., Lacas, F., Veynante, D.: Large eddy simulation and experimental study of flashback and blow-off in a lean partially premixed swirled burner. J. Turb.**5** (2004)
27. Roux, S., Lartigue, G., Poinsot, T., Meier, U., Bérat, C.: Studies of mean and unsteady flow in a swirled combustor using experiments, acoustic analysis and large eddy simulations. Combust. Flame**141** (2005) 40–54
28. Schmitt, P., Poinsot, T.J., Schuermans, B., Geigle, K.: Large-eddy simulation and experimental study of heat transfer, nitric oxide emissions and combustion instability in a swirled turbulent high pressure burner. J. Fluid Mech.**570** (2007) 17–46
29. Kaufmann, A., Nicoud, F., Poinsot, T.: Flow forcing techniques for numerical simulation of combustion instabilities. Combust. Flame**131** (2002) 371–385
30. Truffin, K., Poinsot, T.: Comparison and extension of methods for acoustic identification of burners. Combust. Flame**142**(4) (2005) 388–400
31. Roux, A., Gicquel, L.Y.M., Sommerer, Y., Poinsot, T.J.: Large eddy simulation of mean and oscillating flow in a side-dump ramjet combustor. Combust. Flame(doi:10.1016/j.combustflame.2007.06.015) (2007) in press
32. Boudier, G., Gicquel, L.Y.M., Poinsot, T., Bissières, D., Bérat, C.: Comparison of LES, RANS and experiments in an aeronautical gas turbine combustion chamber. Proc. Combust. Inst.**31** (2007) 3075–3082
33. Schildmacher, K.U., Hoffman, A., Selle, L., Koch, R., Schulz, C., Bauer, H.J., Poinsot, T.: Unsteady flame and flow field interaction of a premixed model gas turbine burner. Proc. Combust. Inst.**31** (2007) 3197–3205
34. Sagaut, P.: Large eddy simulation for incompressible flows. Springer (2002)
35. Mahesh, K., Constantinescu, G., Moin, P.: A numerical method for large-eddy simulation in complex geometries. J. Comput. Phys.**197**(1) (2004) 215–240
36. Mare, F.D., Jones, W.P., Menzies, K.: Large eddy simulation of a model gas turbine combustor. Combust. Flame**137** (2001) 278–295
37. Pitsch, H.: Large eddy simulation of turbulent combustion. Ann. Rev. Fluid Mech.**38** (2006) 453–482
38. El-Asrag, H., Menon, S.: Large eddy simulation of bluff-body stabilized swirling non-premixed flames. Proc. Combust. Inst.**31** (2007) 1747–1754
39. Duwig, C., Fuchs, L., Griebel, P., Siewert, P., Boschek, E.: Study of a confined jurbulent jet: influence of combustion and pressure. AIAA Journal**45**(3) (2007) 624–661
40. Poinsot, T., Candel, S., Trouvé, A.: Application of direct numerical simulation to premixed turbulent combustion. Prog. Energy Comb. Sci.**21** (1996) 531–576
41. Moin, P., Mahesh, K.: Dns: a tool in turbulence research. Ann. Rev. Fluid Mech.**30** (1998) 539–578
42. Vervisch, L., Poinsot, T.: Direct numerical simulation of non premixed turbulent flames. Ann. Rev. Fluid Mech.**30** (1998) 655–692
43. Tennekes, H., Lumley, J.L.: A first course in turbulence. M.I.T. Press, Cambridge (1972)
44. Stoer, J.S., Bulirsch, R.: An introduction to numerical analysis. Springer, Berlin (1980)
45. Chaitin-Chatelin, F., Frayssé, V.: Lectures on Finite Precision Computations. SIAM, Philadelphia (1996)

46. Hanrot, G., Lefèvre, V., Stehlé, D., Zimmermann, P.: Worst cases for a periodic function with large arguments. In Kornerup, P., Muller, J.M., eds.: Proceedings of the 18th IEEE Symposium on Computer Arithmetic, Los Alamitos, CA, IEEE Computer Society Press (2007) 133–140
47. Freitag, M., Janicka, J.: Investigation of a strongly swirled premixed flame using les. Proc. Combust. Inst.**31** (2007) 1477–1485
48. Lele, S.: Compact finite difference schemes with spectral like resolution. J. Comput. Phys.**103** (1992) 16–42
49. Abarbanel, S.S., Chertock, A.E.: Strict stability of high-order compact implicit finite-difference schemes: the role of boundary conditions for hyperbolic PDEs, I. J. Comput. Phys.**160** (2000) 42–66
50. Sengupta, T.K., Ganerwal, G., Dipankar, A.: High accuracy compact schemes and gibbs' phenomenon. J. Sci. Comput.**21**(3) (2004) 253–268
51. Saad, Y.: A flexible inner-outer preconditioned gmres algorithm. SIAM J. Sci. Comput.**14** (1993) 461–469
52. Fraysse, V., Giraud, L., Gratton, S.: A set of flexible-gmres routines for real and complex arithmetics. Technical Report TR/PA/98/20, CERFACS (1998)
53. Cuthill, E., McKee, J.: Reducing the bandwidth of sparse symmetric matrices. In: Proceedings of the 24th National Conference of the ACM. (1969) 157–172
54. Liu, W.H., Sherman, A.H.: Comparative analysis of the cuthill-mckee and the reverse cuthill-mckee ordering algorithms for sparse matrices. SIAM Journal of Numerical Analysis **13**(2) (1976) 198–213
55. Pope, S.B.: Ten questions concerning the large-eddy simulation of turbulent flows. New Journal of Physics **6** (2004) 35
56. Williams, R.D.: Performance of dynamic load balancing algorithms for unstructured mesh calculations. Concurrency: Practice, and Experience **3(5)** (1991) 451–481
57. Taylor, V.E., Nour-Omid, B.: A study of the factorization fill-in for a parallel implementation of the finite element method. Int. J. Numer. Meth. Eng.**37** (1994) 3809–3823
58. Hirsch, C.: Numerical Computation of internal and external flows. John Wiley, New York (1988)
59. Aref, H.: Integrable, chaotic and turbulent vortex motion in two-dimensional flows. Ann. Rev. Fluid Mech.**15** (1983) 345–389