The growth of rounding errors and the repetitivity of Large Eddy Simulation on parallel machines

Jean-Mathieu Senoner[*], Marta García[*], Simon Mendez[*],
Gabriel Staffelbach[†], Olivier Vermorel[†] and Thierry Poinsot[‡]

Abstract

This paper studies the growth rate of rounding errors in LES and shows that instantaneous flow fields produced by LES are partially controlled by these rounding errors and depend on multiple parameters: number of processors used for parallel simulation (even in an explicit code), changes in initial conditions (even of the order of machine accuracy), machine precision (simple, double or quadruple), etc. Using a fully developed turbulent channel flow and a laminar Poiseuille pipe flow as test cases, results show that only turbulent flows exhibit a high sensitivity to these parameters, leading to instantaneous flow fields which can be totally different after a few dozen flow-through times. Even though these results essentially confirm that LES reflects the true nature of turbulence, small perturbations of initial conditions growing rapidly in time, they highlight an often overlooked challenge of LES in terms of validation and prediction of unsteady phenomena.

Introduction

Large Eddy Simulation (LES) has become the most efficient tool to predict non-reacting[1,2] as well as reacting turbulent flows[3–8]. The main strength of LES compared to classical Reynolds Averaged (RANS) methods is that, like Direct Numerical Simulation (DNS)[9–11], 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[12] implies that the flow solution exhibited by LES is very sensitive to any "small perturbations" of a given reference state. These small perturbations 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

[*] PhD student, CERFACS, CFD team.

[†] Senior research fellow, CERFACS, CFD team.

[‡] Research director IMF Toulouse, INP de Toulouse and CNRS. Associate fellow AIAA.

to LES variability. The study of error growth in finite precision computations is an important topic in applied numerical mathematics[13–15] 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 results variability: these conditions are often unknown and any small change in initial conditions may trigger significant changes in the LES solution. Boundary conditions, in particular the unsteady velocity profiles imposed at inlets and outlets, can have the same effect as initial conditions but are not studied here.

- Due to its large computational resource requirements, modern LES heavily relies on parallel computing. Therefore, in codes using domain partitioning, i.e. most of them, this is 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[16], the residual at one node is obtained by adding the weighted residuals of the surrounding cells. Of course, the exact result of this addition is independant of the addition ordering but this not true for rounding errors. Therefore, additions of more than two summands may yield distinct results for floating-point accumulation. For example, 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 summands[17] and after a few tens of thousands iterations, the LES result may be affected. Since the propagation of these rounding errors is 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[2,3,18] or in space such as compact schemes[19–21] is not considered here: for such schemes, the methods[22,23] used to solve the linear system appearing at each iteration depend on the number of processors. Therefore, the propagation of rounding errors is not the only reason why solutions obtained with different numbers of processors vary.

- 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[24,25] may produce the same effect as a simple perturbation and can be the source of solution divergence.

- Finally compilation options, in particular those affecting code optimization and obviously those affecting truncation operations, are a fifth source of LES variability. Different optimization options of the compiler are not tested in the following. Such tests would certainly be of interest although care must be taken since too agressive optimization options can affect scheduling of operations in a physically wrong manner and lead to erroneous results.

The solution of a given LES/DNS at a certain instant unavoidably changes when the rounding errors or the initial conditions are not exactly identical and LES/DNS solutions are known to have a meaning only in a statistical manner[26]. It is however a real difficulty in the practical use of LES/DNS because it means that 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[27], 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 paper tries to address these issues and 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?

The first section gives 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[28]. 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 node 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 in this section. All simulations have been performed on an IBM JS21 supercomputer.

Effects of the number of processors on a fully developed turbulent channel LES

This first example is the LES of a rectangular fully developed turbulent channel of dimensions: 75x25x50 mm (Fig. 1). 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[8]. 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 partitioning is such that the method is perfectly equivalent on any number of processors. The Recursive Inertial Bisection (RIB)[29,30] algorithm has been used to partition the grid and the Cuthill-McKee algorithm is considered as the default node reordering strategy on subdomains. The scheme used here is the Lax-Wendroff scheme[31]. Additional tests were performed using a third-order scheme in space and time[32] but led to the same conclusions.

Figs. 2–4 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, as can be seen in Figs. 5 and 6.

This very simple example illustrates the main question of the present work: is the result of Figs. 2–4 reasonable? If it is not a simple programming error (the next section will show that it is not), can other parameters produce similar effects?

Sensitivity of laminar and turbulent flows to small perturbations

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,
- node ordering,
- time step,
- machine precision (i.e. floating point representation according to IEEE standard).

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:

$$N_{max} = \max(|u_1(\mathbf{x}) - u_2(\mathbf{x})|) \text{ for } x \in \Omega \tag{1}$$

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

where Ω and V_Ω respectively denote the computational domain and its volume. Both norms (in m/s) will be applied to the streamwise 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.

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 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 non determinisitic message arrival is usually implemented in parallel codes to improve performance.

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. Run parameters of the laminar Poiseuille flow are displayed in table 2

Figure 7 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. 2–4, 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} m/s. This is expected since there is only one stable solution for the Poiseuille flow for infinite times and accordingly do laminar flows 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[33] 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. 2–4 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 node 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. 7) at a few points in the flow and grows continuously if the flow is turbulent.

The growth rate of the differences between solutions in simulations TC1 and TC2 cannot be estimated in a simple manner. A simplified description for the determination of growth rates of trajectory separation is found in Jimenez[34], it is briefly summarized in the following. Two-dimensional vortical flows, a description of vortices as points with associated circulations and negligible viscosity are assumed. Under these hypotheses, set of linearized ordinary differential equations can be derived to evaluate the time evolution of the distance between two neighbouring flow field trajectories differing by an arbitrary infinitesimal perturbation $\delta(t)$. This system admits exponential solutions, the growth rates of which are determined by the real part of the eigenvalues. The evolution of inviscid/conservative systems conserves volume in phase space. As the real part of the eigenvalues describes the separation of trajectories in time, it represents a measure of the evolution of the volume in phase space. Thus, the sum of the real parts vanishes and at least one of them has to be positive. At this stage, the number of degrees of freedom of the system imposes topological constraints on the trajectories and can prevent their separation, but a few degrees of freedom suffice for such systems to exhibit chaotic behavior, as demonstrated by the famous Lorenz attractor[35]. This argument illustrates that the separation of trajectories is a property related to the nature of vorticity and mainly driven by the number of degrees of freedom in phase space. Therefore, a simple estimate of the growth rate from flow parameters does a priori not seem possible. Although the simplifications in the described analysis are severe, one can suppose that independently of the spatial distribution and amplitude (within the limit of the linearity assumption) of perturbations applied to a given turbulent flow field, the separation of trajectories for various simulations yields similar exponential growth rates, which is confirmed in the following. Moreover, it is a purely physical phenomenon and though induced by rounding errors, the growth rate should not depend on numerical parameters such as machine precision or time step. These aspects are adressed in forthcoming sections.

Influence of initial conditions

The previous section has shown that turbulence combined with a different domain partitioning (i.e. a different number of processors) 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 partitioning. This is verified in runs TC3 and TC4 which are run on a single processor, 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 single 10^{-16} m/s perturbation is applied to the streamwise velocity component. Simulations with different locations of the perturbation were run to ensure that the position did not affect results.

Figure 8 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. 8): 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 the accumulation of rounding errors along the interface between subdomains at each time step while TC3 and TC4 differ only through the initial condition: the sequence of floating point operations is exactly the same in both cases. 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 different sequences of rounding errors or to a perturbation of the initial conditions.

Effects of node ordering in the mesh

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 focuses 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 demonstrate. TC5 and TC3 are performed with the same initial condition and run on one processor only. The only difference is the node reordering strategy.

As shown by Fig. 9, 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 a modified code will probably never yield the same instantaneous flow fields, potentially leading to discussions on the validity of the modifications.

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. 10 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. 10 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 CFL number.[31]

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. 11 and compared to the difference between TC1 and TC2.

Figure 11 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 identical difference levels for all three cases. This shows that higher precision computations cannot prevent the exponentional divergence of trajectories but only delay it.

Conclusions

This work focused on the sensitivity of instantaneous LES fields to different rounding error propagation due to situations where various parameters of the run such as number of processors, initial condition, time step, changes in addition ordering of cell residuals for cell vertex methods are modified. The baseline simulation used for the tests was a fully developed turbulent channel. The conclusions are the following:

- 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 different propagation of rounding errorss rounding errors induced by domain partitioning and scheduling of operations. More generally, any change in the code lines affecting the propagation of rounding errors will have the same effects.

- Small changes in initial condition (of the order of machine accuracy at one point of the flow only) produce similar 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[26] in which solutions are only meaningful in a statistical sense and instantaneous values cannot be used for analysis. However, on a more practical level, they point out various difficulties to develop LES codes:

- Repeating a given LES after modifying the code and verifying that instantaneous solutions have not changed is not always possible. Since any programming error will affect instantaneous solutions in the same way, identifying errors introduced by new lines will require a detailed analysis based on statistically averaged fields and a significant amount of time.

- Validating an 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.

- Porting a LES code from one machine to another will also produce different instantaneous solutions for turbulent runs, making comparisons and validations of new architectures difficult.

More generally, these results demonstrate that the concept of "quality" in LES will require much more detailed studies and tools than what has been used up to now in Reynolds Averaged simulations.

Acknowledgments

We thank Prof. F. Chaitin-Chatelin for helpful discussions.
The first author gratefully acknowledges the funding by the European Community in the framework of the Marie Curie Early Stage Research Training Fellowship (contract number MEST-CT-2005-020426).

References

[1] Sagaut, P., *Large eddy simulation for incompressible flows*, Springer, 2002.

[2] Mahesh, K., Constantinescu, G., and Moin, P., "A numerical method for large-eddy simulation in complex geometries," *Journal of Computational Physics* , Vol. 197, No. 1, 2004, pp. 215–240.

[3] Di Mare, F., Jones, W. P., and Menzies, K., "Large Eddy Simulation of a model gas turbine combustor," *Combustion and Flame* , Vol. 137, 2001, pp. 278–295.

[4] Poinsot, T. and Veynante, D., *Theoretical and numerical combustion*, R.T. Edwards, 2nd edition., 2005.

[5] Pitsch, H., "Large Eddy Simulation of Turbulent Combustion," *Annual Review of Fluid Mechanics* , Vol. 38, 2006, pp. 453–482.

[6] El-Asrag, H. and Menon, S., "Large Eddy Simulation of bluff-body stabilized swirling non-premixed flames," *Proceedings of the Combustion Institute*, Vol. 31, 2007, pp. 1747–1754.

[7] Duwig, C., Fuchs, L., Griebel, P., Siewert, P., and Boschek, E., "Study of a Confined Turbulent Jet: Influence of Combustion and Pressure," *AIAA Journal* , Vol. 45, No. 3, March 2007, pp. 624–661.

[8] Schmitt, P., Poinsot, T., Schuermans, B., and Geigle, K., "Large-eddy simulation and experimental study of heat transfer, nitric oxide emissions and combustion instability in a swirled turbulent high pressure burner," *Journal of Fluid Mechanics* , Vol. 570, 2007, pp. 17–46.

[9] Poinsot, T., Candel, S., and Trouvé, A., "Application of direct numerical simulation to premixed turbulent combustion," *Progress in Energy and Combustion Science* , Vol. 21, 1996, pp. 531–576.

[10] Moin, P. and Mahesh, K., "DNS: a tool in turbulence research," *Annual Review of Fluid Mechanics* , Vol. 30, 1998, pp. 539–578.

[11] Vervisch, L. and Poinsot, T., "Direct Numerical Simulation of non premixed turbulent flames," *Annual Review of Fluid Mechanics* , Vol. 30, 1998, pp. 655–692.

[12] Tennekes, H. and Lumley, J., *A first course in turbulence*, M.I.T. Press, Cambridge, 1972.

[13] Stoer, J. and Bulirsch, R., *An introduction to numerical analysis*, Springer, Berlin, 1980.

[14] Chaitin-Chatelin, F. and Frayssé, V., *Lectures on Finite Precision Computations*, SIAM, Philadelphia, 1996.

[15] Overton, M. L., *Numerical Computing with IEEE Floating Point Arithmetic*, SIAM, Philadelphia, 2001.

[16] Schönfeld, T. and Rudgyard, M., "Steady and Unsteady Flows Simulations Using the Hybrid Flow Solver AVBP," *AIAA Journal* , Vol. 37, No. 11, 1999, pp. 1378–1385.

[17] Hanrot, G., Lefèvre, V., Stehlé, D., and Zimmermann, P., "Worst cases for a periodic function with large arguments," *Proceedings of the 18th IEEE Symposium on Computer Arithmetic*, edited by P. Kornerup and J.-M. Muller, IEEE Computer Society Press, Los Alamitos, CA, 2007, pp. 133–140.

[18] Freitag, M. and Janicka, J., "Investigation of a strongly swirled premixed flame using LES," *Proceedings of the Combustion Institute*, Vol. 31, 2007, pp. 1477–1485.

[19] Lele, S., "Compact finite difference schemes with spectral like resolution," *Journal of Computational Physics* , Vol. 103, 1992, pp. 16–42.

[20] Abarbanel, S. S. and Chertock, A. E., "Strict stability of high-order compact schemes: the role of boundary conditions for hyperbolic PDEs, I," *Journal of Computational Physics* , Vol. 160, 2000, pp. 42–66.

[21] Sengupta, T. K., Ganerwal, G., and Dipankar, A., "High accuracy compact schemes and Gibbs' phenomenon," *Journal of Scientific Computing*, Vol. 21, No. 3, 2004, pp. 253–268.

[22] Saad, Y., "A flexible inner-outer preconditioned GMRES algorithm," *SIAM J. Sci. Comput.*, Vol. 14, 1993, pp. 461–469.

[23] Fraysse, V., Giraud, L., and Gratton, S., "A set of Flexible-GMRES routines for real and complex arithmetics," Tech. Rep. TR/PA/98/20, CERFACS, 1998.

[24] Cuthill, E. and McKee, J., "Reducing the bandwidth of sparse symmetric matrices," *Proceedings of the 24th National Conference of the ACM*, 1969, pp. 157–172.

[25] Liu, W.-H. and Sherman, A. H., "Comparative analysis of the Cuthill-Mckee and the reverse Cuthill-Mckee ordering algorithms for sparse matrices," *SIAM Journal of Numerical Analysis*, Vol. 13, No. 2, 1976, pp. 198–213.

[26] Pope, S. B., "Ten questions concerning the large-eddy simulation of turbulent flows," *New Journal of Physics*, Vol. 6, 2004, pp. 35.

[27] Sommerer, Y., Galley, D., Poinsot, T., Ducruix, S., Lacas, F., and Veynante, D., "Large eddy simulation and experimental study of flashback and blow-off in a lean partially premixed swirled burner," *Journal of Turbulence*, Vol. 5, 2004.

[28] Moureau, V., Lartigue, G., Sommerer, Y., Angelberger, C., Colin, O., and Poinsot, T., "High-order methods for DNS and LES of compressible multi-component reacting flows on fixed and moving grids," *Journal of Computational Physics* , Vol. 202, No. 2, 2005, pp. 710–736.

[29] Williams, R. D., “Performance of dynamic load balancing algorithms for unstructured mesh calculations,” *Concurrency: Practice, and Experience*, Vol. 3(5), 1991, pp. 451–481.

[30] Taylor, V. E. and Nour-Omid, B., “A Study of the Factorization Fill-in for a Parallel Implementation of the Finite Element Method,” *International Journal for Numerical Methods in Engineering*, Vol. 37, 1994, pp. 3809–3823.

[31] Hirsch, C., *Numerical Computation of internal and external flows*, John Wiley, New York, 1988.

[32] Colin, O. and Rudgyard, M., “Development of high-order Taylor-Galerkin schemes for unsteady calculations,” *Journal of Computational Physics* , Vol. 162, No. 2, 2000, pp. 338–371.

[33] Aref, H., “Integrable, chaotic and turbulent vortex motion in two-dimensional flows,” *Annual Review of Fluid Mechanics* , Vol. 15, 1983, pp. 345–389.

[34] Jimenez, J., “Turbulence and vortex dynamics,” Tech. Rep. Ecole Polytechnique, 2004.

[35] Lorenz, E. N., “Deterministic Nonperiodic Flow,” *Journal of the Atmospheric Sciences*, Vol. 20, 1963, pp. 130–141.

List of Tables

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

Run	Nbr	Init.	Precision	Graph	CFL
Id	proc	cond.		ordering	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

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

List of Figures

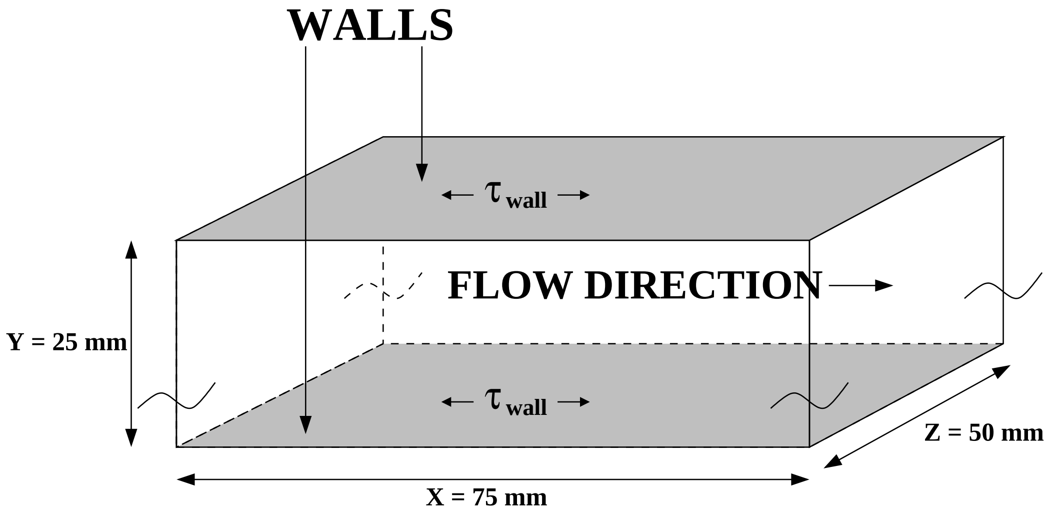
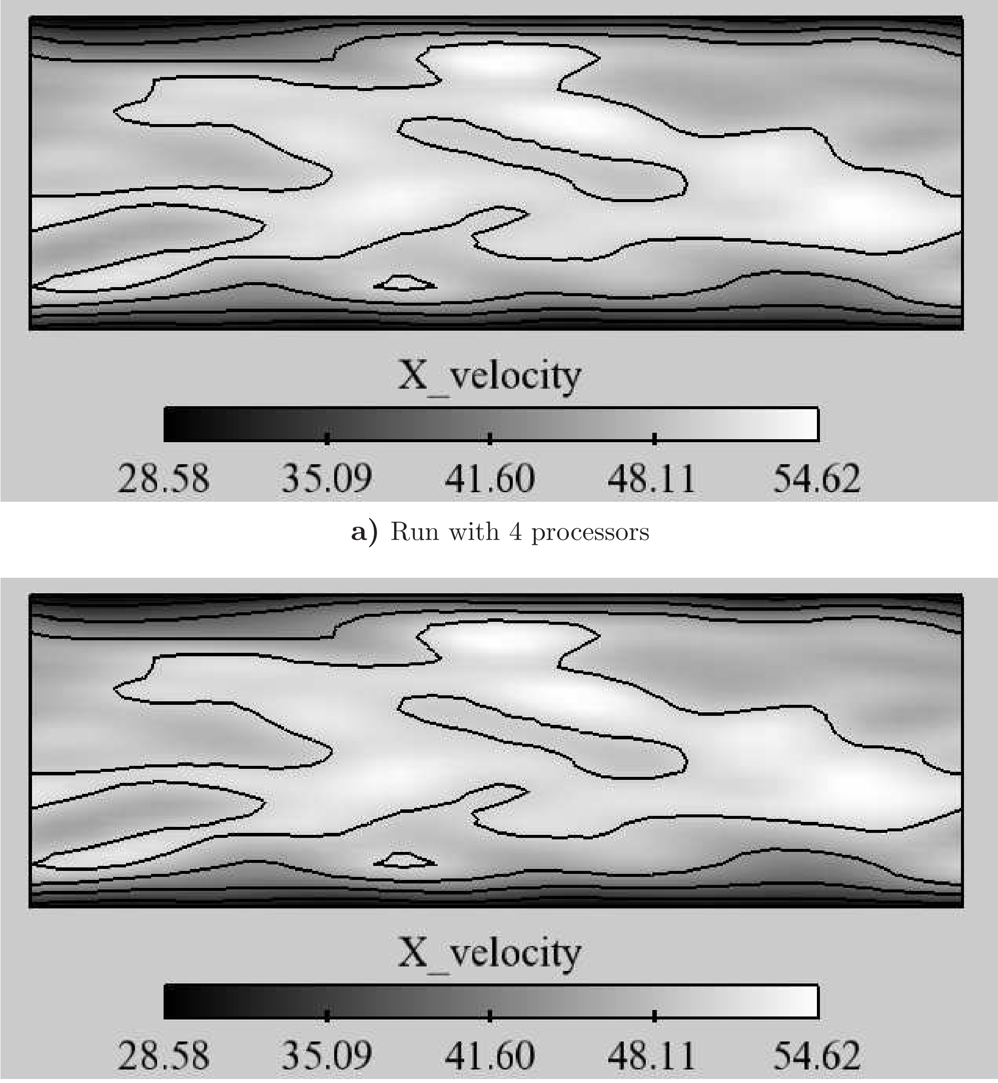


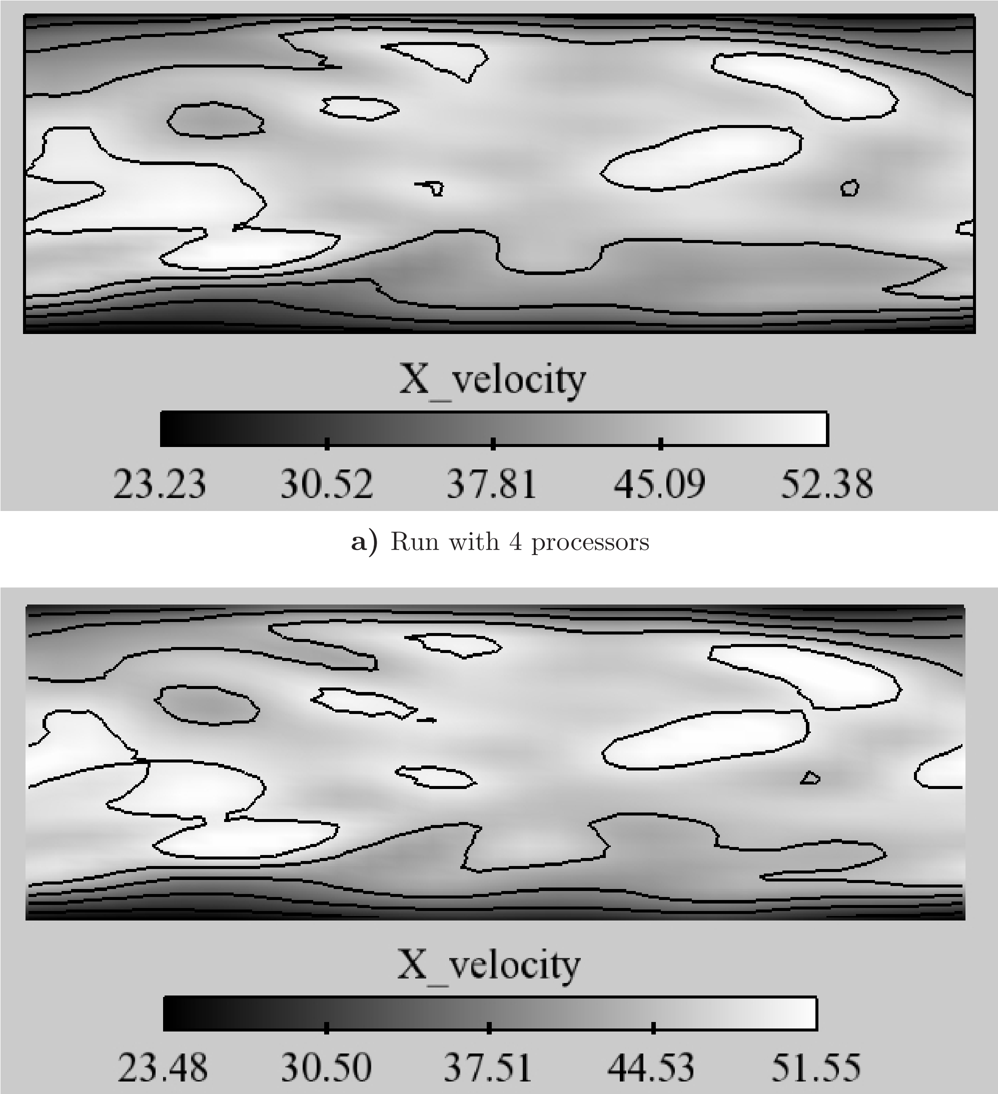
Figure 1: Schematic of a periodic channel. The upper and lower boundaries consist of walls, all other boundaries are pairwise periodic.



a) Run with 4 processors

b) Run with 8 processors

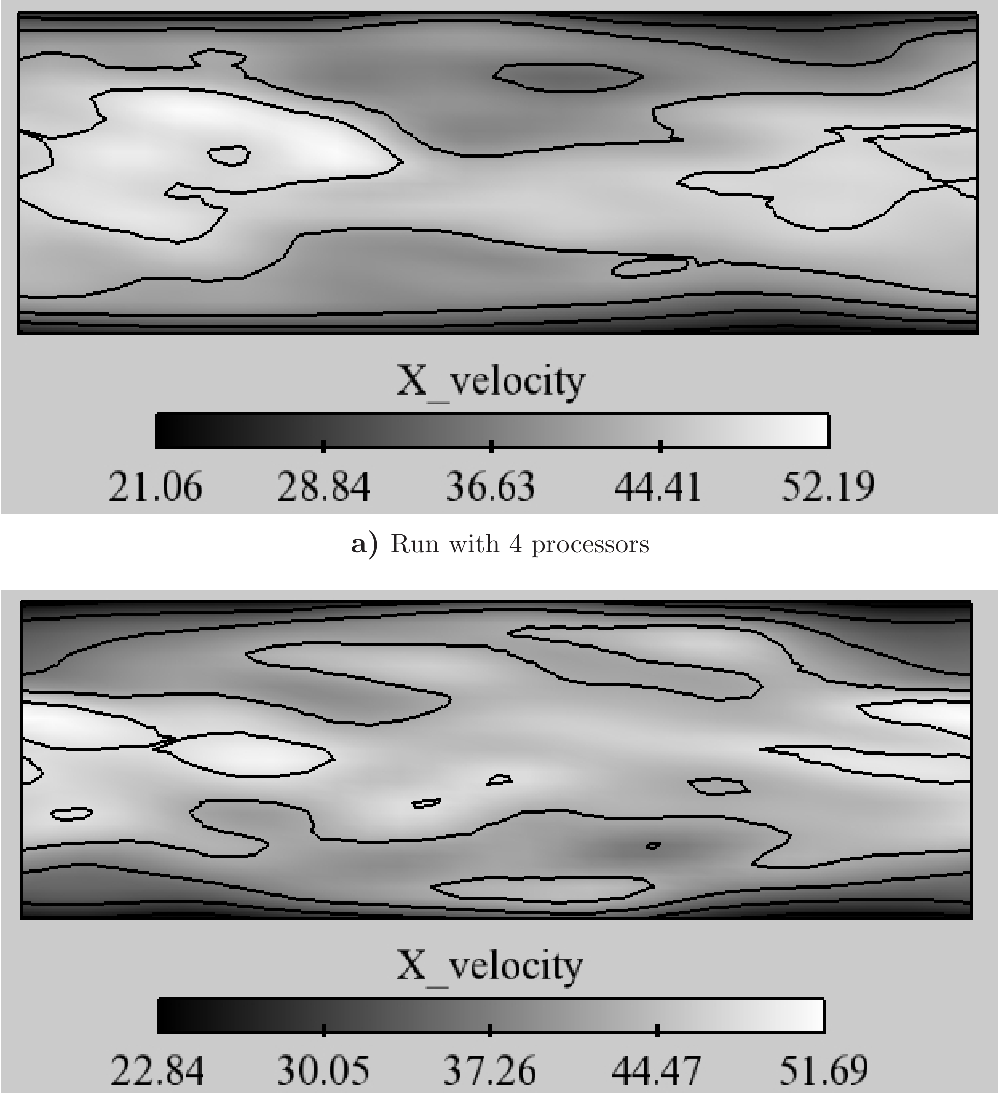
Figure 2: Instantaneous field of axial velocity in the central plane of the channel at $t+ = 7.68$**. a) run TC1 (4 processors), b) run TC2 (8 processors).**



a) Run with 4 processors

b) Run with 8 processors

Figure 3: 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

Figure 4: 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).**

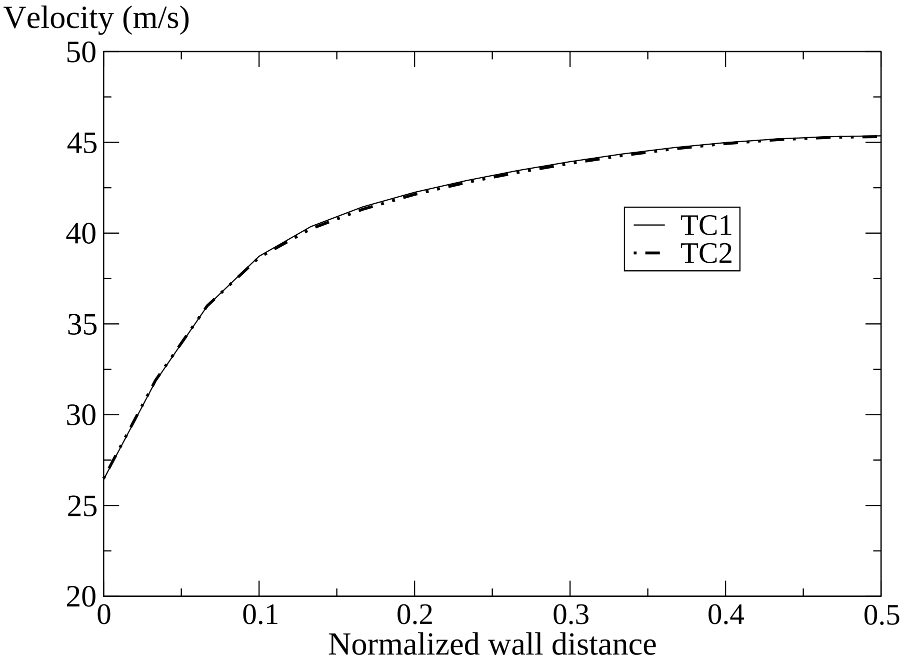


Figure 5: Comparison of the mean velocity profiles for TC1 (4 processors) and TC2 (8 processors) over half channel height.

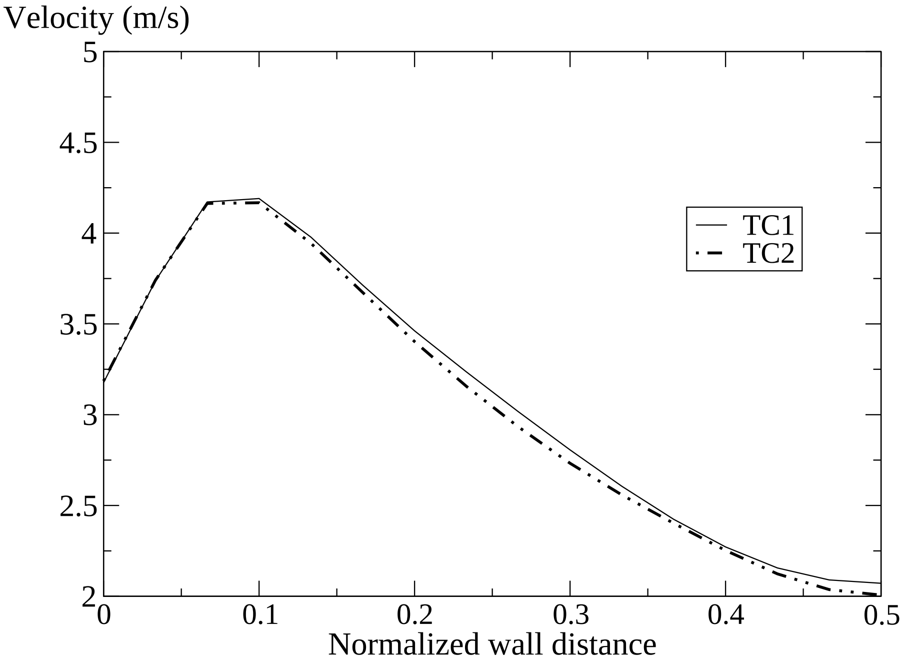


Figure 6: **Comparison of the root mean square velocity profiles for TC1 (4 processors) and TC2 (8 processors) over half channel height.**

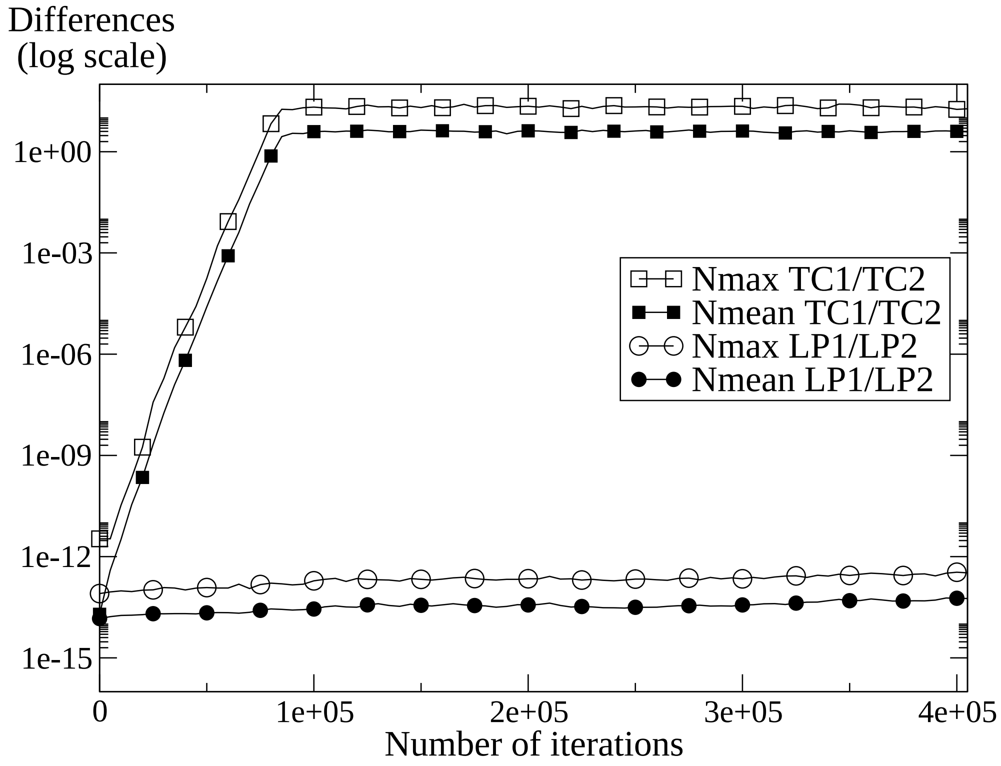


Figure 7: Effects of turbulence. Differences between solutions measured by N_{max} (open symbols) and N_{mean} (closed symbols) versus iteration. Squares: differences between TC1 and TC2 (turbulent channel). Circles: differences between LP1 and LP2 (laminar Poiseuille flow).

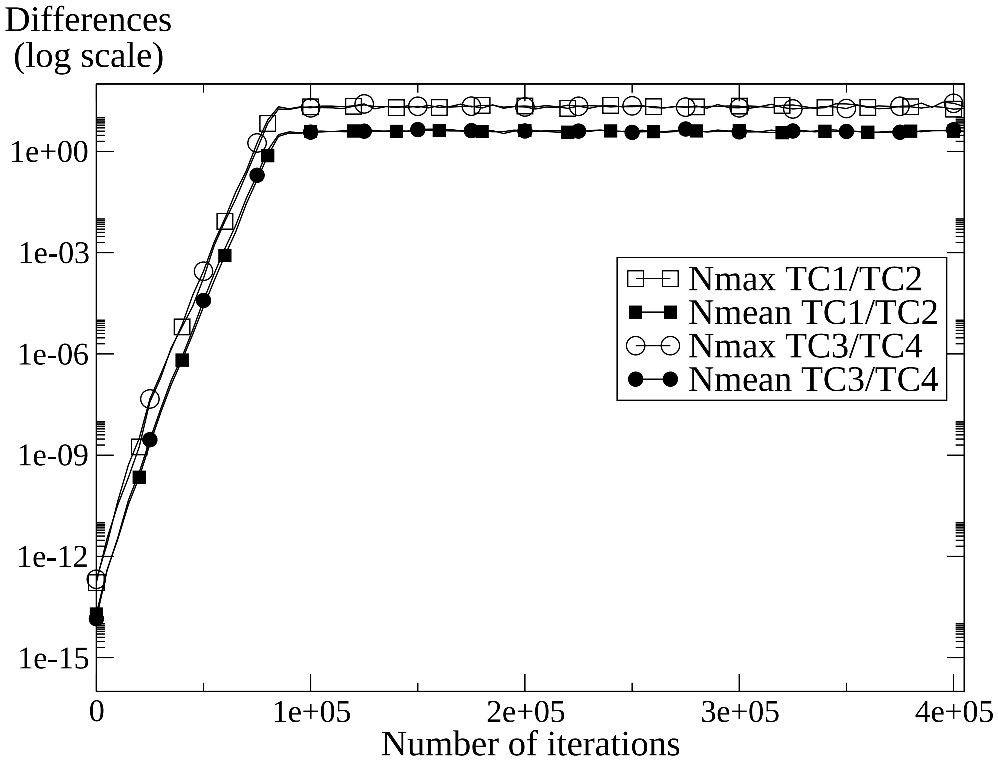


Figure 8: **Effects of initial conditions. Differences between solutions measured by N_{max} (open symbols) and N_{mean} (closed symbols) versus iteration. Squares: differences between TC1 and TC2 (different numbers of processors). Circles: differences between TC3 and TC4 (different initial conditions).**

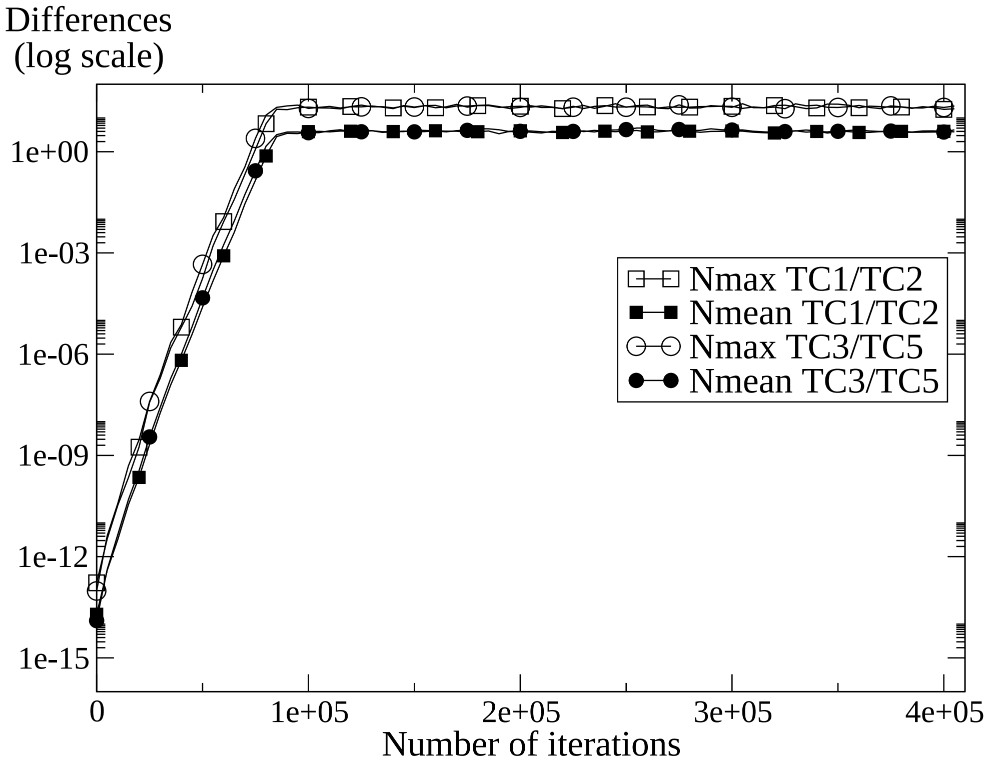


Figure 9: Effects of addition order. Differences between solutions measured by N_{max} (open symbols) and N_{mean} (closed symbols) versus iteration. Squares: differences between TC1 and TC2. Circles: differences between TC3 and TC5.

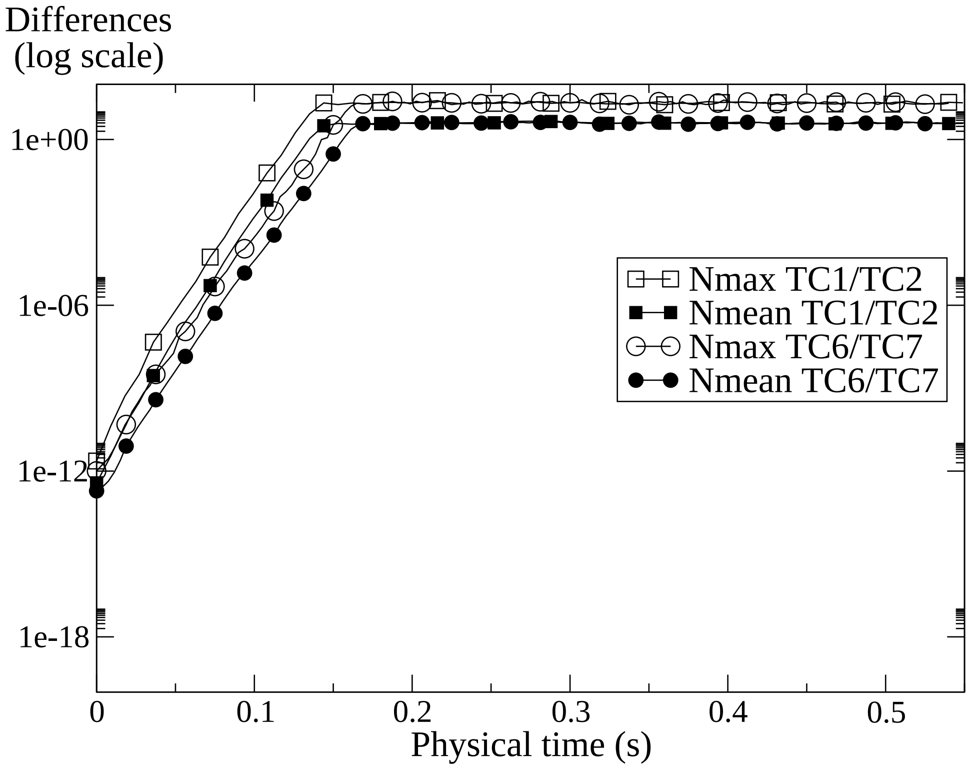


Figure 10: Effects of time step. Differences between solutions measured by N_{max} (open symbols) and N_{mean} (closed symbols) versus physical time. Squares: differences between TC1 and TC2 (time step Δt). Circles: differences between TC6 and TC7 (time step $\Delta t/2$).

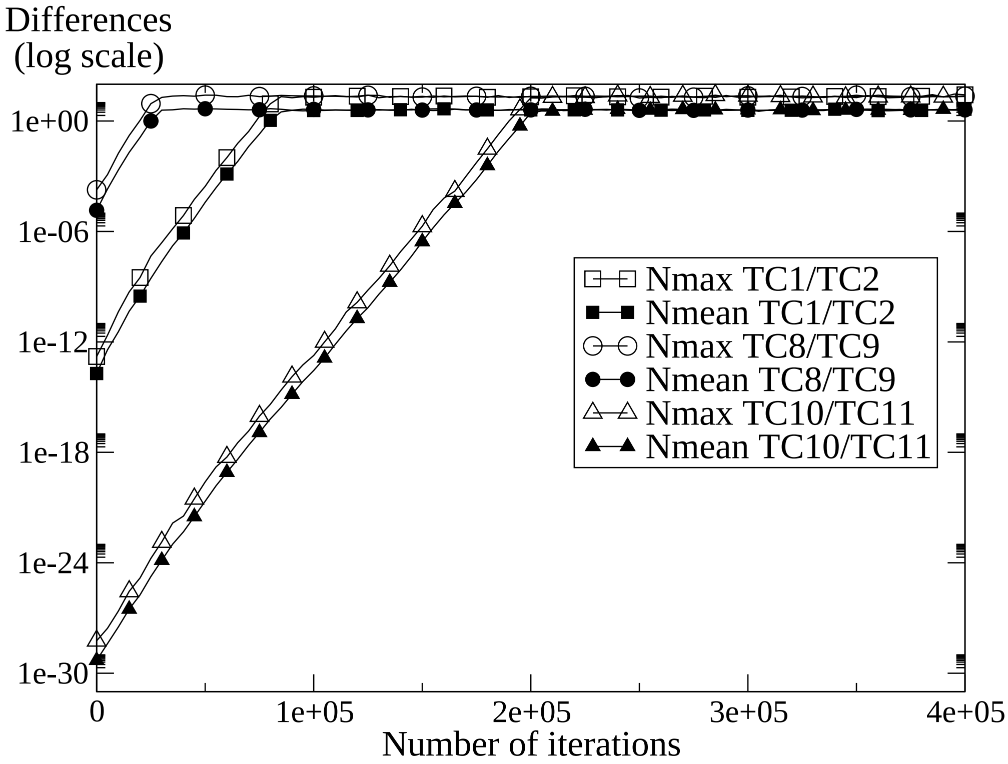


Figure 11: Effects of machine accuracy. Differences between solutions measured by N_{max} (open symbols) and N_{mean} (closed symbols) versus iteration. Squares: differences between TC1 and TC2 (double precision). Circles: differences between TC8 and TC9 (simple precision). Triangles: differences between TC10 and TC11 (quadruple precision)