Large-Eddy Simulations of Turbulent Flows, from Desktop to Supercomputer (Invited Talk)

Ugo Piomelli[1], Alberto Scotti[2], and Elias Balaras[3]

[1] University of Maryland, College Park MD 20742, USA,
ugo@eng.umd.edu,
WWW: http://www.glue.umd.edu/~ugo
[2] University of North Carolina, Chapel Hill NC 27599-3300, USA
scotti@marine.unc.edu
[3] University of Maryland, College Park MD 20742, USA,
balaras@eng.umd.edu

Abstract. In this paper, a general introduction to the large-eddy simulation (LES) technique will be given. Modeling and numerical issues that are under study will be described to illustrate the capabilities and requirements of this techniques. A palette of applications will then be presented, chosen on the basis both of their scientific and technological importance, and to highlight the application of LES on a range of machines, with widely different computational capabilities.

1 Introduction

Turbulent flows are ubiquitous in nature and in technological applications. They occur in such diverse fields as metereology, astrophysics, aerospace, mechanical, chemical and environmental engineering. For this reason, turbulence has been the object of study for many centuries. In 1510, Leonardo da Vinci accompanied a drawing of the vortices shed behind a blunt obstacle (Fig. 1) with the following observation:

> Observe the motion of the water surface, which resembles that of hair, that has two motions: one due to the weight of the shaft, the other to the shape of the curls; thus, water has eddying motions, one part of which is due to the principal current, the other to the random and reverse motion.

Despite its importance, and the number of researchers that have studied it theoretically, experimentally and, recently, numerically, turbulence remains one of the open problems in Mechanics.

The equations that govern turbulent flows are the Navier-Stokes equations. For turbulent flows, no exact solutions are available, and their numerical solution is made difficult by the fact that an accurate calculation depends critically on the accurate representation, in space and time, of the coherent fluid structures (eddies) that govern to a very large extent the transfer of momentum and energy. The direct solution of the Navier-Stokes equations (also known as "direct

Fig. 1. Sketch from Leonardo da Vinci's notebooks.

numerical simulation", or DNS) is an extremely expensive endeavor in turbulent flows. Its cost depends on the cube of the Reynolds number, the dimensionless parameter that measures the relative importance of convective and diffusive effects. At present, DNS calculations are limited to flows with Reynolds numbers $O(10^4)$, while most engineering and geophysical applications are characterized by $Re = O(10^6 - 10^9)$.

Practical, predictive, calculations require the use of simplified models. The most commonly used one is the solution of the Reynolds-averaged Navier-Stokes equations (RANS), in which the flow variables are decomposed into a mean and a fluctuating part, as fore-shadowed in da Vinci's observations, and the effect of the turbulent eddies is parameterized globally, through some more-or-less complex turbulence model. This technique is widespread in industrial practice, but turbulence models are found to require *ad hoc* adjustments from one flow to another, due to the strongly flow-dependent nature of the largest eddies, which contribute most to the energy and momentum transfer, and which depend to a very significant extent on the boundary conditions. Furthermore, they fail to give any information on the wavenumber and frequency distribution of the turbulent eddies, which may be important in acoustics, or in problems involving the interaction of fluid with solid structures.

The large-eddy simulation (LES) is a technique intermediate between DNS and RANS, which relies on computing accurately the dynamics of the large eddies while modeling the small, subgrid scales of motion. This method is based on the consideration that, while the large eddies are flow-dependent, the small scales tend to be more universal, as well as isotropic. Furthermore, they react more rapidly to perturbations, and recover equilibrium quickly. Thus, the modelling of the subgrid scales is significantly simpler than that of the large scales, and can be more accurate.

Despite the fact that the small scales are modeled, LES remains a fairly computationally intensive technique. Since the motion of the large scales must

be computed accurately in time and space, fine grids (or high-order schemes) and small time-steps are required. Since the turbulent motions are intrinsically three-dimensional (3D), even flows that are two- or one-dimensional in the mean must be computed using a 3D approach. Finally, to accumulate the averaged statistics needed for the engineering design and analysis, the equations of motion must be integrated for long times.

As a result of these computational requirements, until recently LES has been a research tool, used mostly in academic environments and research laboratories to study the physics of turbulence. Most calculations were carried out on vector machines (Cray X-MP, Y-MP and C90, for instance). Typical computations of flows at moderate Reynolds number required up to 1 million grid points, and used times of the order of 100 CPU hours and more on such machines.

Recently, progress has been made on two fronts. First, the development of advanced models [4, 5] for the small-scale contribution to momentum transfer, the subgrid-scale stresses, allows the accurate prediction of the response of the small scales even in non-equilibrium situations. Secondly, the decreasing cost of computational power has made it possible to perform larger simulations on a day-to-day basis, even using inexpensive desktop workstations. Simulations using 3 million grid points can easily be run on Pentium-based computers. The turn-around time for a mixing-layer simulation that used 5 million points on a dedicated Alpha processor is of the order of two days per integral scale of the flow, a time comparable to what was achievable on a Cray, in which the greater processor speed was often offset by the load of the machine, and the end-user was frequently restricted to a few CPU hours per day. With the increased availability of inexpensive workstation clusters, the application of LES is bound to become more and more affordable. The use of large, massively parallel computers is, however, still required by very advanced, complex applications that may require very large numbers of grid points [$O(10^7)$], and correspondingly long integration times.

In this article, a general introduction to LES will be given. Although particular emphasis will be placed on numerical issues, the main thrust of the paper will not be the algorithmic problems and developments, but rather a discussion of the capabilities and computational requirements of this techniques. A palette of applications will then be presented, chosen on the basis both of their scientific and technological importance, and to highlight the application of LES on a range of machines, with widely different computational capabilities. This article should not be seen as a comprehensive review of the area; the reader interested in more in-depth discussions of the subject is addressed to several recent reviews [1–3].

2 Governing equations

The range of scales present in a turbulent flow is a strong function of the Reynolds number. Consider for instance the mixing layer shown in Fig. 2. The largest eddies in this flow are the spanwise rollers, whose scale is L; a very wide range of smaller scales is present. The energy supplied to the largest turbulent eddies

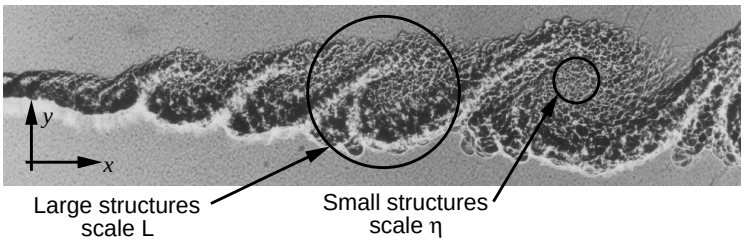

Fig. 2. Visualization of the flow in a mixing layer (from Brown & Roshko [6]). The flow is from left to right; a splitter plate (immediately to the left of the image) separates a high-speed flow (top) from a low-speed one. The two streams then mix, forming the large, quasi-2D rollers in the figure, as well as a range of smaller scales.

by the mean flow is transferred to smaller and smaller scales (energy cascade), and eventually dissipated into heat by the smallest ones. Most of the energy, in fact, is dissipated by eddies contained in a length scale band of about 6η to 60η, where η is the so-called Kolmogorov scale.

In DNS, all the scales of motion, up to and including the dissipative scales of order η must be resolved; since the computational domain must be significantly larger than the large scale L, while the grid size must be of order η, the number of grid points required is proportional to the ratio L/η. It can be shown that this ratio is proportional to $Re^{3/4}$, where the Reynolds number $Re = \Delta U L/\nu$ is based on the velocity difference between the two streams, ΔU, and an integral scale of the flow, L; ν is the kinematic viscosity of the fluid. Thus, the number of grid points needed to perform a three-dimensional DNS scales like the 9/4 power of the Reynolds number.

The time-scale of the smallest eddies also supplies a bound for the maximum time-step allowed: since the ratio of the integral time-scale of the flow to the Kolmogorov time-scale is also proportional to $Re^{3/4}$ the number of time-steps required to advance the solution by a fixed time has the same dependence on Re. Assuming that the CPU time required by a numerical algorithm is proportional to the total number of points N, the cost of a calculation will depend on the product of the number of points by the number of time-steps, hence to Re^3.

In an LES only the large scales of motion must be resolved. The similarity of the small scales, which only transmit energy to smaller scales, and the fact that the global dissipation level is set by the large scales (even though the dissipation takes place at the small-scale level) are exploited by SGS models, whose main purpose is to reproduce the energy transfer accurately, at least in a statistical sense. When the filter cutoff is in the inertial region of the spectrum (i.e., in the wave-number range in which the energy cascade takes place), therefore, the resolution required by an LES is nearly independent of the Reynolds number.

In wall-bounded flows, in which the scale of the large, energy-carrying eddies is Reynolds-number-dependent, the situation is less favorable. The cost of an LES is still, however, significantly reduced over that of a DNS.

To separate the large from the small scales, LES is based on the definition of a filtering operation: a filtered (or resolved, or large-scale) variable, denoted by an overbar, is defined as

$$\overline{f}(\mathbf{x}) = \int_D f(\mathbf{x}')G(\mathbf{x},\mathbf{x}';\overline{\Delta})d\mathbf{x}', \tag{1}$$

where D is the entire domain, G is the *filter* function, and $\overline{\Delta}$, the filter width, is a parameter that determines the size of the largest eddy removed by the filtering operation. The filter function determines the size and structure of the small scales. It is easy to show that, if G is a function of $\mathbf{x}-\mathbf{x}'$ only, differentiation and the filtering operation commute [7].

The most commonly-used filter functions are the sharp Fourier cutoff filter, best defined in wave space[1]

$$\widehat{G}(k) = \begin{cases} 1 \text{ if } k \le \pi/\overline{\Delta} \\ 0 \text{ otherwise,} \end{cases} \tag{2}$$

the Gaussian filter,

$$G(x) = \sqrt{\frac{6}{\pi\overline{\Delta}^2}}\exp\left(-\frac{6x^2}{\overline{\Delta}^2}\right), \tag{3}$$

and the top-hat filter in real space:

$$G(x) = \begin{cases} 1/\overline{\Delta} \text{ if } |x| \le \overline{\Delta}/2 \\ 0 \quad \text{otherwise,} \end{cases} \tag{4}$$

For uniform filter width $\overline{\Delta}$ the filters above are mean-preserving and commute with differentiation.

The effect of filtering a test function with increasing filter-width is shown in Fig. 3. Although an increasing range of small scales is removed as $\overline{\Delta}$ is increased, the large-scale structure of the signal is preserved. In RANS, on the other hand, the effect of all turbulent eddies would be removed by the averaging procedure.

In LES the filtering operation (1) is applied formally to the governing equations; this results in the filtered equations of motion, which are solved in LES. For an incompressible flow of a Newtonian fluid, they take the following form:

$$\frac{\partial\overline{u}_i}{\partial x_i} = 0. \tag{5}$$

$$\frac{\partial\overline{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\overline{u}_i\overline{u}_j\right) = -\frac{1}{\rho}\frac{\partial\overline{p}}{\partial x_i} - \frac{\partial\tau_{ij}}{\partial x_j} + \nu\frac{\partial^2\overline{u}_i}{\partial x_j\partial x_j}. \tag{6}$$

The filtered Navier-Stokes equations written above govern the evolution of the large, energy-carrying, scales of motion. The effect of the small scales appears through a subgrid-scale (SGS) stress term,

$$\tau_{ij} = \overline{u_iu_j} - \overline{u}_i\overline{u}_j, \tag{7}$$

that must be modeled to achieve closure of the system of equations.

[1] A quantity denoted by a caret $\widehat{\cdot}$ is the complex Fourier coefficient of the original quantity.

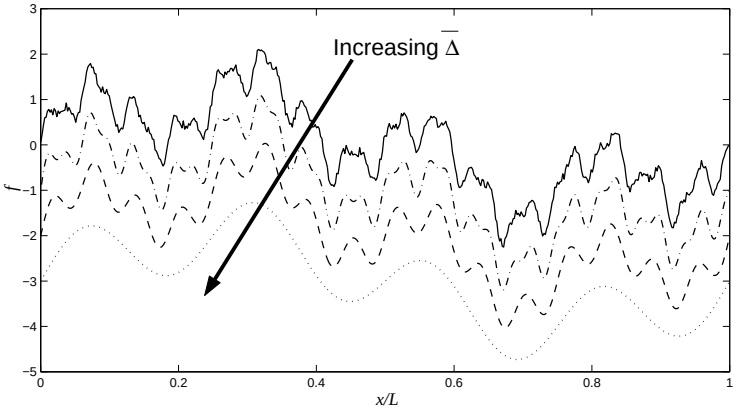

Fig. 3. Effect of filtering a test function with increasing filter-width $\overline{\Delta}$.

3 Subgrid-scale models

In LES the dissipative scales of motion are resolved poorly, or not at all. The main role of the subgrid-scale model is, therefore, to remove energy from the resolved scales, mimicking the drain that is associated with the energy cascade. Most subgrid scale models are eddy-viscosity models of the form

$$\tau_{ij} - \frac{\delta_{ij}}{3}\tau_{kk} = -2\nu_T \overline{S}_{ij}, \tag{8}$$

that relate the subgrid-scale stresses τ_{ij} to the large-scale strain-rate tensor $\overline{S}_{ij} = (\partial\overline{u}_i/\partial x_j + \partial\overline{u}_j/\partial x_i)/2$. In most cases the equilibrium assumption (namely, that the small scales are in equilibrium, and dissipate entirely and instantaneously all the energy they receive from the resolved ones) is made to simplify the problem further and obtain an algebraic model for the eddy viscosity [8]:

$$\nu_T = C\overline{\Delta}^2|\overline{S}|\overline{S}_{ij}; \qquad |\overline{S}| = (2\overline{S}_{ij}\overline{S}_{ij})^{1/2}. \tag{9}$$

This model is known as the "Smagorinsky model". The value of the coefficient C can be determined from isotropic turbulence decay [9]; if the cutoff in the inertial subrange, the Smagorinsky constant $C_s = \sqrt{C}$ takes values between 0.18 and 0.23 (and $C \simeq 0.032 - 0.053$). In the presence of shear, near solid boundaries or in transitional follows, however, it has been found that C must be decreased. This has been accomplished by various types of *ad hoc* corrections such as van Driest damping [10] or intermittency functions [11].

More advanced models, that do not suffer from the shortcomings of the Smagorinsky model (excessive dissipation, incorrect asymptotic behavior near solid surfaces, need to adjust the constant in regions of laminar flow or high shear) have been developed recently. The introduction of dynamic modeling

ideas [4] has spurred significant progress in the subgrid-scale modeling of non-equilibrium flows. In dynamic models the coefficient(s) of the model are determined as the calculation progresses, based on the energy content of the smallest resolved scale, rather than input *a priori* as in the standard Smagorinsky [8] model. A modification of this model was proposed by Meneveau *et al.* [5], which has been shown to give accurate results in non-equilibrium flows in which other models fail [12].

Turbulence theory (in particular the Eddy-Damped Quasi-Normal Markovian theory) has also been successful in aiding the development of SGS models. The Chollet-Lesieur [13, 14] model, as well as the structure-function [15] and filtered-structure-function models [16] have been applied with some success to several flows.

A detailed discussion of SGS models is beyond the scope of this paper. The interested reader is referred to the review articles referenced above [1–3].

4 Numerical methods

In large-eddy simulations the governing equations (5-6) are discretized and solved numerically. Although only the large scales of motion are resolved, the range of scales present is still significant. In this section, a brief overview of the numerical requirements of LES will be given.

4.1 Time advancement

The choice of the time advancement method is usually determined by the requirements that numerical stability be assured, and that the turbulent motions be accurately resolved in time. Two stability limits apply to large-eddy simulations. The first is the viscous condition, that requires that the time-step Δt be less than $\Delta t_v = \sigma \Delta y^2/\nu$ (where σ depends on the time advancement chosen). The CFL condition requires that Δt be less than $\Delta t_c = \mathrm{CFL}\Delta x/u$, where the maximum allowable Courant number CFL also depends on the numerical scheme used. Finally, the physical constraint requires Δt to be less than the time scale of the smallest resolved scale of motion, $\tau \sim \Delta x/U_c$ (where U_c is a convective velocity of the same order as the outer velocity).

In many cases (especially in wall-bounded flows, and at low Reynolds numbers), the viscous condition demands a much smaller time-step than the other two; for this reason, the diffusive terms of the governing equations are often advanced using implicit schemes (typically, the second-order Crank-Nicolson scheme). Since, however, Δt_c and τ are of the same order of magnitude, the convective term can be advanced by explicit schemes such as the second-order Adams-Bashforth method, or third- or fourth-order Runge-Kutta schemes.

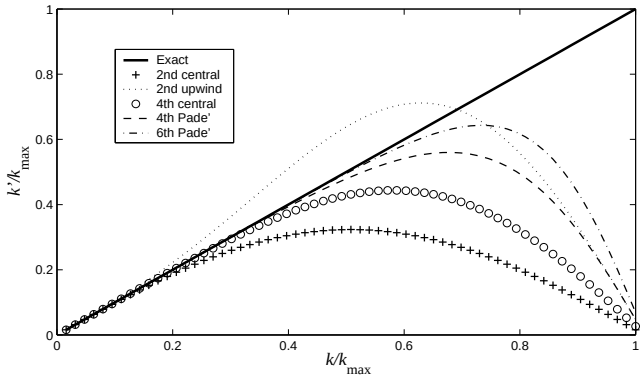

Fig. 4. Modified wave-number for various differencing schemes.

4.2 Spatial discretization

The analytical derivative of a complex exponential $f(x) = e^{ikx}$ is $f'(x) = ike^{ikx}$; if f is differentiated numerically, however, the result is

$$\frac{\delta f}{\delta x} = ik'e^{ikx}, \tag{10}$$

where k' is the "modified wave-number". A modified wave-number corresponds to each differencing scheme. Its real part represents the attenuation of the computed derivative compared to the actual one, whereas a non-zero imaginary part of k' indicates that phase errors are introduced by the numerical differentiation. Figure 4 shows the real part of the modified wave-numbers for various schemes. For a second-order centered scheme, for instance, $k' = k\sin(k\Delta x)/(k\Delta x)$. For small wave-numbers k the numerical derivative is quite accurate; high wave-number fluctuations, however, are resolved poorly. No phase errors are introduced.

The need to resolve accurately high wave-number turbulent fluctuations implies that either low-order schemes are used on very fine meshes (such that, for the smallest scales that are physically important, $k' \simeq k$), or that higher-order schemes are employed on coarser meshes. High-order schemes are more expensive, in terms of computational resources, than low-order ones, but the increase in accuracy they afford (for a given mesh) often justifies their use.

4.3 Conservation

It is particularly important, in large-eddy simulations of transitional and turbulent flows, that the numerical scheme preserves the conservation properties of the Navier-Stokes equations. In the limit $Re \to \infty$, the Navier-Stokes equations conserve mass, momentum, energy and vorticity in the interior of the flow: the integral of these quantities over the computational domain can only be affected

through the boundaries. Some numerical schemes, however, do not preserve this property. For instance, the convective term in the momentum equations can be cast in several ways:

$$\text{Advective form} : u_j \frac{\partial u_i}{\partial x_j}, \tag{11}$$

$$\text{Divergence form} : \frac{\partial}{\partial x_j}(u_i u_j), \tag{12}$$

$$\text{Rotational form} : \epsilon_{ijk} u_j \omega_k - \frac{\partial}{\partial x_i}(u_j u_j/2), \tag{13}$$

$$\text{Skew-symmetric form} : \frac{1}{2}\left[u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial}{\partial x_j}(u_i u_j)\right], \tag{14}$$

where $\omega_k = \epsilon_{kij}\partial u_j/\partial x_i$. It is easy to show (Morinishi *et al.* [17]) that, if a typical co-located finite-difference scheme is used, the first form does not conserve either momentum or energy, the second conserves momentum but not energy, the others conserve both. If, on the other hand, a control-volume approach is used, the divergence form conserves energy but the pressure-gradient term does not. With a staggered grid, the divergence form preserves the conservation properties of the Navier-Stokes equations if central, second-order accurate differences are used.

Upwind schemes also have very undesirable effects on the conservation properties of the calculation, as does the explicit addition of artificial dissipation. Even mildly upwind-biased schemes result in a significant loss of accuracy. These methods are not suited to LES of incompressible flows, and should be avoided.

4.4 Complex geometries

For applications to complex geometries, single-block, Cartesian meshes are inadequate, since they do not give the required flexibility. One alternative is the use of body-fitted curvilinear grids. LES codes in generalized coordinates have been used, among others by Zang *et al.* [21, 22] (who applied it to a Cartesian geometry, the lid-driven cavity [21], and to the study of coastal up-welling [22, 23]), Beaudan and Moin [24] and Jordan [25]. Jordan [25] examined the issue of filtering in curvilinear coordinates, and concluded that filtering the transformed (in the generalized coordinates) equations directly in the computational space is better than performing the filtering either of the transformed equations in real space, or of the untransformed equations in Cartesian space.

Even if curvilinear grids are used, the application of LES to complex geometries might be limited by resolution requirements. In the presence of a solid boundary, for instance, a very fine mesh is required to resolve the wall layer. Kravchenko *et al.* [26] used zonal embedded meshes and a numerical method based on B-splines to compute the flow in a two-dimensional channel, and around a circular cylinder. The use of the B-splines allows use of an arbitrarily high order of accuracy for the differentiation, and accurate interpolation at the interface between the zones. A typical grid for the channel flow simulations is shown in

Fig. 5, which evidences the different spanwise resolution in the various layers, in addition to the traditional stretching in the wall-normal direction. The use of zonal grids allowed Kravchenko *et al.* [26] to increase the Reynolds number of the calculations substantially: they performed an LES of the flow at $Re_c = 109\,410$ using 9 embedded zones allowed them to resolve the wall-layer using a total of 2 million points. A single-zone mesh with the same resolution would have under-resolved the wall layer severely. The mean velocity profile was in excellent agreement with the experimental data.

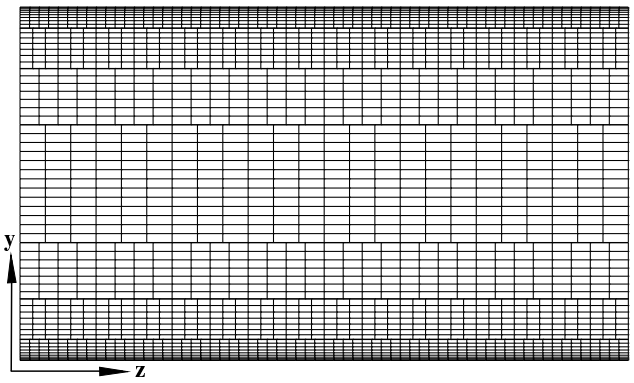

Fig. 5. Zonal embedded grid with fine grid zones near the walls and coarse zones in the middle of the channel. The flow is into the paper. Reproduced with permission from Kravchenko *et al.* [26]

Very few applications of LES on unstructured meshes have been reported to date. Jansen [29] showed results for isotropic turbulence and plane channel. For the plane channel, the results were in fair agreement with DNS data (the peak streamwise turbulence intensity, for instance, was 15% higher than that obtained in the DNS), but slightly better than the results of finite-difference calculations on the same mesh. Simulations of the flow over a low-Reynolds number airfoil using this method [28] were in fair agreement with experimental data. Knight *et al.* [30] computed isotropic turbulence decay using tetrahedral meshes, and compared the Smagorinsky model with results obtained relying on the numerical dissipation to drain energy from the large scales. They found that the inclusion of an SGS model gave improved results.

While high-order schemes can be applied fairly easily in simple geometries, in complex configurations their use is rather difficult. Present applications of LES to relatively complex flows, therefore, tend to use second-order schemes; the increasing use of LES on body-fitted grids for applications to flows of engineering interest, indicates that, at least in the immediate future, second-order accurate schemes are going to increase their popularity, at the expense of the spectral methods that have been used frequently in the past. Explicit filtering of the

governing equations, with filter widths larger than the grid size may be required in such circumstances.

5 Applications: flow in an accelerating boundary layer

A boundary layer is the region of fluid flow nearest to a solid body, in which viscous effects (i.e., diffusion) are important. Turbulent boundary layers occur in many technological applications, and are often subjected to favorable pressure gradients that result in an acceleration of the velocity at the edge of the boundary layer, the free-stream velocity. Figure 5 illustrates schematically the boundary layer that occurs at the leading edge of an airplane wing. The fluid is accelerated as it turns over the top side of the airfoil from the stagnation point, where its velocity is zero.

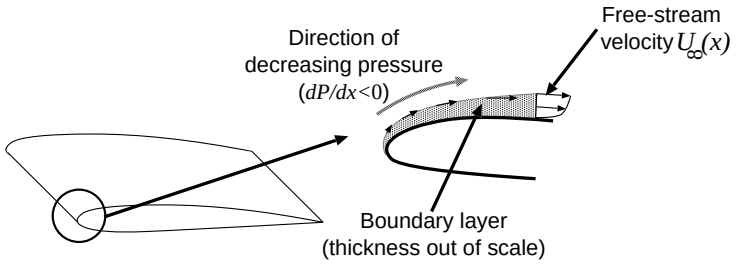

Fig. 6. Sketch of the flow near the leading edge of an airfoil.

Despite the importance of this type of flow fields, however, they are not as well understood as the canonical zero-pressure-gradient boundary layer, due to the much wider parameter space, and to the difficulty in determining universal scaling laws similar to those for the zero-pressure-gradient case. In fact, a large percentage of the investigations of accelerating flows to date have concentrated on self-similar cases, in which such scaling laws can be found.

It is recognized that, if the acceleration is sufficiently strong, turbulence cannot be sustained. In self-similar accelerating boundary layer, this phenomenon takes place when the acceleration parameter K reaches a critical value:

$$K = \frac{\nu}{U_\infty^2}\frac{dU_\infty}{dx} = -\frac{\nu}{\rho U_\infty^3}\frac{dP_\infty}{dx} \simeq 3 \times 10^{-6}. \tag{15}$$

The RANS approach, which is often used in aeronautical applications, has difficulty dealing with reversion of a turbulent flow to a laminar one, and into the re-transition of the flow, that becomes turbulent again as the acceleration ceases on the suction (upper) side of the airfoil. Large-eddy simulation can help in understanding the physics that cause reversion and re-transition, as well as

provide accurate data that can be used for the development and validation of lower-level RANS models to be used in engineering design.

In particular, experimental evidence indicates that the dynamics of the coherent eddies play an important role in the reversion. An improved understanding of the dynamics of these eddies in boundary layers subjected to a favorable pressure gradient would be extremely beneficial. Apart from the considerations about momentum transfer and mixing also valid in other flows, an additional motivating factor is provided here by the consideration that most of the theoretical attempts to derive scaling laws are often based on multiple-scale approximations that assume little or no interaction between inner and outer layers. The most direct way to establish the validity of this assumption is by studying the coherent eddies in the wall layer. Unlike RANS solutions, in which only the average flow-field is computed, LES can supply information on the behavior of the coherent structures.

Piomelli and co-workers [31] studied the velocity fields obtained from the large-eddy simulation (LES) of accelerating boundary layers with the aim to improve the understanding of the dynamics of the coherent vortices in the re-laminarizing flows. To separate the effect of the pressure gradient from that of curvature, the calculation of the boundary layer on a flat plate with an accelerating free-stream was carried out; the configuration is similar to the flow on the lower wall of a wind-tunnel in which the upper wall converges, as sketched in Fig. 7. The computational domain is the shaded area in the figure.

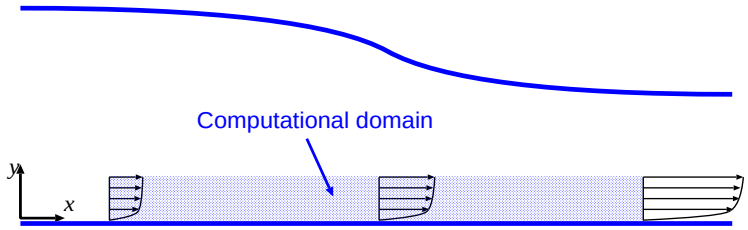

Fig. 7. Sketch of the physical configuration. Accelerating boundary layer.

Two computations were examined: one in which the acceleration is relatively mild (the maximum velocity increases by 35% over the computational domain, and $K < 3 \times 10^{-6}$ everywhere), and a strong-acceleration case in which the velocity increases by almost 150%, and $K > 3 \times 10^{-6}$ for a significant portion of the flow. The modification of the turbulence structure in accelerating flows was emphasized, and it was shown how the acceleration can be associated to lower turbulence levels and to the dynamics of the quasi-streamwise coherent vortices.

5.1 Numerical method

The governing equations(5–6) are integrated numerically using the fractional time-step method [18, 19], in which first the Helmholtz equation is solved to ob-

tain an estimate of the velocity field that does not satisfy mass conservation; the pressure is then computed by solving Poisson's equation, the estimated velocity field supplying the source term. When a pressure correction is applied, the resulting velocity will be a divergence-free solution of the Navier-Stokes equations. If the Navier-Stokes equations are written as

$$\frac{\partial \overline{u}_i}{\partial t} = -\frac{\partial \overline{p}}{\partial x_i} - H_i + \nu \nabla^2 \overline{u}_i, \tag{16}$$

where H_i contains the nonlinear term and the SGS stresses, the time-advancement sequence based on the second-order-accurate Adams-Bashforth method consists of the following steps:

1. Velocity prediction (Helmholtz equation):

$$v_j - u_j^n = \Delta t \left[\frac{3}{2}(-H_j^n + \nabla^2 \overline{u}_j{}^n) - \frac{1}{2}(-H_j^{n-1} + \nabla^2 \overline{u}_j{}^{n-1}) \right]; \tag{17}$$

2. Poisson solution:

$$\nabla^2 \overline{p} = \frac{1}{\Delta t} \frac{\partial v_j}{\partial x_j}; \tag{18}$$

3. Velocity correction:

$$\overline{u}_j{}^{n+1} = v_j - \Delta t \frac{\partial \overline{p}}{\partial x_j}; \tag{19}$$

v_j is the estimated velocity. This time-advancement scheme is second-order-accurate in time. The code uses central differences on a staggered mesh, and is second-order accurate in space as well. Discretization of the Poisson equation (18) results in an hepta-diagonal matrix that can be solved directly if the grid is uniform in at least one direction.

The calculations were performed on a domain of size $400 \times 25 \times 25$. All lengths are normalized with respect to the inflow displacement thickness δ_o^*; the displacement thickness is an integral length scale defined as

$$\delta^* = \int_0^\infty \left(1 - \frac{U}{U_\infty}\right) dy, \tag{20}$$

where U is the average streamwise velocity. The calculations used $256 \times 48 \times 64$ grid points. A grid-refinement study was performed in the strong-acceleration case, in which the number of grid points was increased by 50% in each direction. In the accelerating-flow region ($x/\delta_o^* < 320$) the results on the coarser mesh matched very well those obtained with the finer one. In the re-transitioning area, the qualitative behaviour of the flow was captured correctly, but some differences (of the order of 15%) were observed in the statistical quantities. The Lagrangian dynamic eddy viscosity model [5] was used to parameterize the SGS stresses.

The cost of the computations, was 2.2×10^{-5} CPU seconds per time-step and grid point on a 300 MHz Pentium II running Linux. Out of this CPU time,

37% is devoted to the computation of the RHS, 25% to the computation of the turbulent viscosity, 12% to solve the Poisson equation, and 10% to update the velocity field and impose boundary conditions. The rest of the CPU is consumed by I/O and computation of statistical quantities. Typically a computation on a 10^6 grid requires approximately 42 hours of CPU to obtain converged statistics (sampling over 1.5 flow-through times). It is interesting to observe that the cost of solving the Poisson equation is a small fraction of the total cost when a direct solver (as in the present case) is used. Any other choice of solution method, like multigrid methods, conjugate gradient methods, etc. would substantially increase the cost of this step, which can account for a large fraction of the total cost, depending on the problem and the computational grid.

5.2 Results

The free-stream velocity obtained from the calculation, U_∞, the pressure parameter K and the momentum-thickness Reynolds number, $Re_\theta = \theta U_\infty/\nu$, where θ is the momentum thickness defined as

$$\theta = \int_0^\infty \left(1 - \frac{U}{U_\infty}\right) \frac{U}{U_\infty} dy, \tag{21}$$

are shown in Fig. 8 for the two cases examined. In the strong acceleration case, despite the presence of a fairly extended region in which K exceeds 3×10^{-6}, the Reynolds number never goes below the critical value $Re_\theta \simeq 350$. Thus one would expect the flow to become less turbulent, but not to revert fully into a laminar one.

The streamwise development of several time-averaged and integral quantities is shown in Fig. 9. As a result of the free-stream acceleration, the boundary layer becomes thinner, as shown by the distributions of δ^* and θ. The skin friction coefficient based on the local free-stream velocity, $C_f = 2\tau_w/\rho U_\infty^2$ (where τ_w is the wall stress), initially increases, but, as the flow begins to relaminarize, it decreases in both the mild- and strong-acceleration case.

Although the pressure-gradient parameter K is well above the critical value of 3×10^{-6} in the strongly accelerating case, the acceleration is not sustained long enough for the Reynolds number to be reduced below the critical value, $Re_\theta \simeq 350$. Thus, full relaminarization does not occur; the shape factor H only reaches a value of 1.6 (the shape factor associated with the laminar Falkner-Skan similarity profile for sink flows of this type is 2.24). The mean velocity profile, however, is significantly affected by the acceleration, even in the mild acceleration case.

As the flow is gradually accelerated, the turbulence adjusts to the perturbation; the turbulent quantities, however, lag the mean flow. The turbulent kinetic energy, for instance, increases in absolute levels, although not as fast as the kinetic energy of the mean flow. Thus, the contours of the turbulent kinetic energy normalized by the kinetic energy of the mean flow, shown in Fig. 10, highlight a significant drop in the turbulent kinetic energy in the region of acceleration.

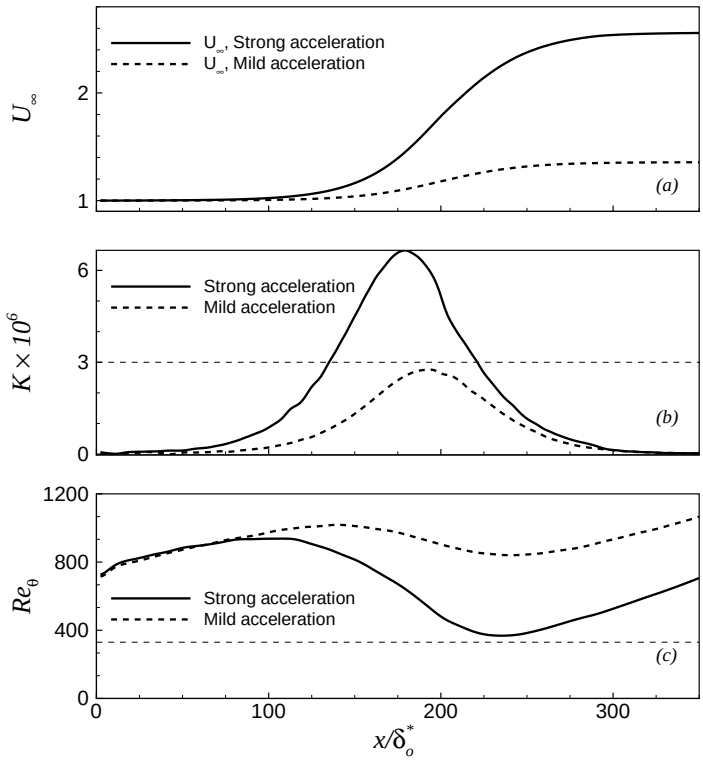

Fig. 8. Spatial development of the free-stream velocity U_∞, the acceleration parameter K, and the momentum-thickness Reynolds number Re_θ in the accelerating boundary layer.

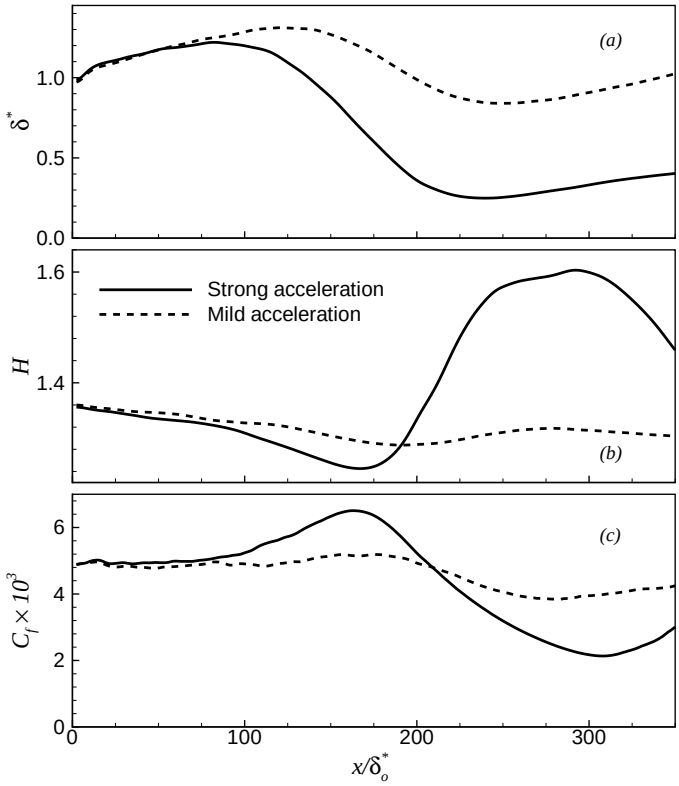

Fig. 9. Spatial development of mean quantities in the accelerating boundary layer. *(a)* Displacement thickness δ^*; *(b)* shape factor H; *(c)* skin-friction coefficient C_f.

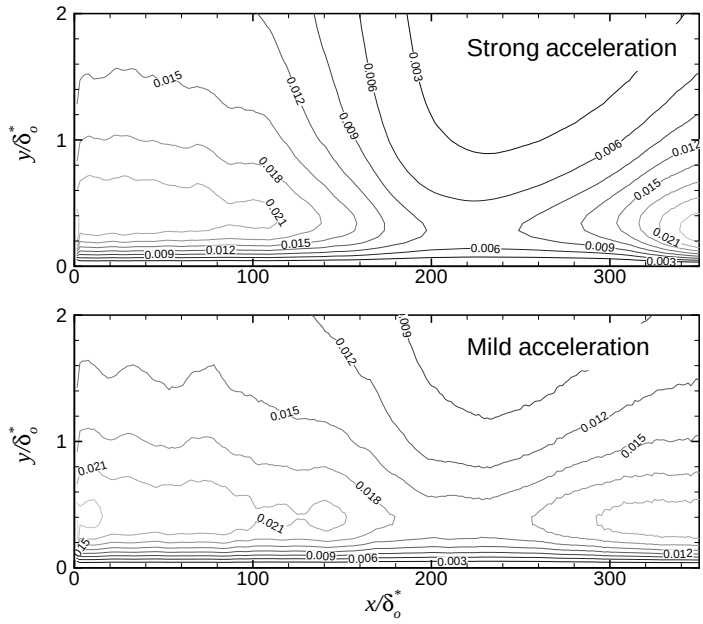

Fig. 10. Contours of the turbulent kinetic energy, normalized by the free-stream kinetic energy in the accelerating boundary layer.

Paradoxically, in many turbulent flows, whenever energy is added through the mean flow, the energy of the turbulence initially decreases, as the coherent vortices adapt to the perturbation. This process often involves disruption of the vortical structures prior to their re-generation. Such is the case in this configuration as well: the vortical structures are visualized in Fig. 11 as isosurfaces of the second invariant of the velocity-gradient tensor, Q, a useful quantity to visualize the regions of high rotation that correspond to the coherent vortices. In the zero-pressure-gradient region near the inflow (top picture) many vortices can be observed, and they are roughly aligned with the flow direction, but form an angle to the wall. This picture is typical of zero-pressure-gradient boundary layers. In the accelerating region, on the other hand, fewer eddies are observed, and those present are more elongated and more closely aligned in the streamwise direction. This structure can be explained based on the fact that the mean velocity gradient has the effect of stretching and re-orienting the vortices. As they are stretched, their vorticity is increased by conservation of angular momentum, while their radius is decreased. The smaller, more intense eddies thus generated are more susceptible to be dissipated by viscous effects.

This calculation highlights a significant advantage of LES over lower-level models. Whenever the coherent eddies play an important role in the flow evolution, RANS calculations (in which the effect of all turbulent eddies is averaged out) cannot predict the flow development accurately. LES, on the other hand,

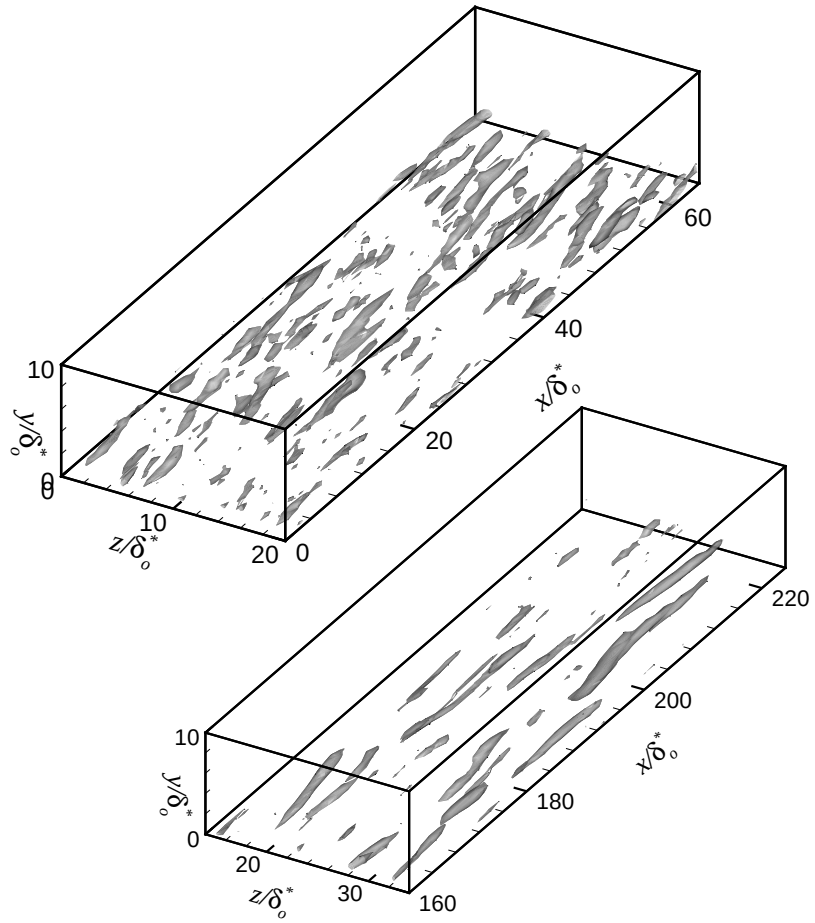

Fig. 11. Instantaneous iso-surfaces of $Q(\delta^*_o/U_o)^2 = 0.02$ in the strong-acceleration case. Top: zero-pressure-gradient region. Bottom: acceleration region.

has a better chance of following the dynamics of the coherent structures, as well as their response to the imposed perturbations.

6 Applications: flow in an oscillating channel

6.1 Motivation

Inherent unsteadiness of the driving conditions characterizes many turbulent flows, both natural (*e.g.* the gravity wave induced in ocean-bottom boundary layers, the blood flow in large arteries, the flow of air in lungs) and artificial (such as the flow in the intake of a combustion engine or the flow in certain heat exchangers). The characterization of unsteady boundary layers is crucial to many disciplines, such as the study of sediment transport in coastal waters, the biology of blood circulation, and so on; moreover, as was pointed out by Sarpkaya [32], by looking at features that are common to steady and unsteady boundary layers, we may better understand the underlying physics of turbulent flows altogether. As already recognized by Binder *et al.* [33], there are no special technical difficulties in performing DNS of pulsating flows. On the other hand, the same authors point out that the oscillating nature of the forcing is felt by the small scales too, so that before trusting the outcome of a LES based on standard closures, a careful (*a posteriori*) comparison with DNS has to be done. This is particularly true for eddy viscosity models, which rely on the combined assumptions that the SGS stress tensor τ_{ij} is aligned with the rate of strain and that the eddy viscosity is proportional to the magnitude of the stress. The latter postulate is somewhat relaxed for the dynamic Smagorinsky model of Germano *et al.* [4], since the eddy viscosity depends on the flux of energy towards the subgrid scales.

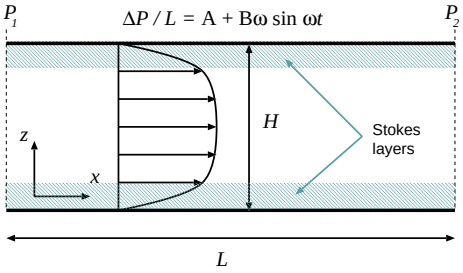

Fig. 12. Sketch of the physical configuration. Oscillating channel flow.

To study the response of turbulence to an oscillating mean flow, a plane-channel flow driven by an oscillating pressure gradient was studied. The physical configuration is illustrated in Fig. 12: the flow between two flat plates that extend to $\pm\infty$ in the streamwise (x) and spanwise (y) directions is simulated.

To drive this periodic flow, a pressure gradient per unit length is introduced on the right-hand-side of the Navier-Stokes equations as a source term. In the case under investigation, this pressure gradient is given by $1 \times 10^{-4} + \omega \sin \omega t$, where ω is the angular frequency of the oscillation. This is the kind of flow considered by Binder *et al.* [33]. The flow admits a laminar solution, which is a trivial extension of the Stokes problem. The flow first decelerates (as it is subjected to the adverse pressure gradient during the first half of the cycle), then accelerates again. During the acceleration phase, as observed before, the flow tends to relaminarize, whereas the adverse-pressure-gradient has the opposite effect, and makes the flow more turbulent.

Since the core of the flow, where the velocity is large, is dominated by convective effects, while the regions near the solid boundary, where the velocity gradients are significant, are dominated by diffusive effects, there is a disparity in time-scales between these two regions: the diffusive time-scale being smaller than the convective one by orders of magnitude. Thus, as the frequency is changed, one would expect a significantly different coupling between the near-wall region (the inner layer) and the core of the flow (the outer layer). To study this coupling, calculations were carried out for a range of frequencies.

Although the geometry is rather simple, and the grids used relatively coarse, this calculation still requires a large amount of CPU time. This is due to the long integration time necessary to achieve convergence. Since phase-averaged data is required, between eight and ten cycles of the oscillation are needed to obtain converged statistical samples. If the frequency is low, the equations of motion must be integrated for very long integration.

6.2 Numerical method

The starting point for this calculation is a well-known serial spectral code for the solution of the filtered Navier-Stokes equation in a channel geometry [34, 35]. Fourier expansions are used in the homogeneous (horizontal) directions, while Chebychev collocation is used in the vertical direction. The code is very highly optimized for a vector architecture. Time-advancement is performed using the fractional time-step method described above; however, the implicit Crank-Nicolson method is used for the vertical diffusion and a low-storage third-order Runge-Kutta scheme is employed for the remaining terms. The procedure described in Section 5.1 still applies, with few obvious modifications. Each sub-step of the Runge-Kutta scheme follows the sequence:

1. Compute the nonlinear terms H_i^n and the horizontal diffusive terms $\nu \nabla^2 \overline{u}_i{}^n$ with information at time t_n. Both these terms are computed in real space.
2. Transform the right-hand side $H^n + \nu \nabla^2 \overline{u}_i{}^n$ into Fourier space.
3. Update the predicted solution in Fourier space:

$$\left(1 - \frac{\nu \Delta t}{2} \frac{\partial^2}{\partial x_3^2}\right) \widehat{v}_j = \left(1 + \frac{\nu \Delta t}{2} \frac{\partial^2}{\partial x_3^2}\right) \widehat{\overline{u}}_j^n + \Delta t \widehat{H}^n \tag{22}$$

by solving implicitly the vertical diffusive problem. Since a Chebychev collocation method is used in the vertical direction z a full matrix obtains for

each mode, which is inverted iteratively by a Generalized Minimum Residual method.

4. Solve the Poisson problem for the pressure in Fourier space:

$$\left(\frac{\partial^2}{\partial z^2} - k_x^2 - k_y^2\right)\widehat{\overline{p}} = -\frac{1}{\Delta t}\left[i(k_1\widehat{v}_1 + k_2\widehat{v}_2) + \frac{\partial \widehat{v}_3}{\partial x_3}\right] \tag{23}$$

again using a Generalized Minimum Residual method to solve the system of linear equations.

5. Update the solution:

$$\widehat{\overline{u}}_1^{n+1} = \widehat{v}_1 - ik_1\widehat{\overline{p}}; \qquad \widehat{\overline{u}}_2^{n+1} = \widehat{v}_2 - ik_2\widehat{\overline{p}}; \qquad \widehat{\overline{u}}_3^{n+1} = \widehat{v}_3 - \Delta t\frac{\partial \widehat{\overline{p}}}{\partial x_3}. \tag{24}$$

An initial series of numerical experiments was performed on an SGI Origin 2000 with 32 R10000 processors running at 195 MHz, each equipped with 640 Mb of Ram and 4Mb of cache, owned by the University of North Carolina. Each processor is rated at 390 MFLOPS. Four discretizations were chosen, $32\times32\times32$, $64\times64\times64$, $128\times128\times96$ and $128\times192\times128$. The serial code experience a drop in performance as the domain grows, from 60 MFLOPS to about 19. This is due to the limited cache, which acts as a bottleneck. The problem is made more acute by the fact that the discrete Fourier transform, which is the heart of the code, is a nonlocal operation. Frigo and Johnson [36] performed extensive testing of different FFT routines on cache based machines, and, without exception, all routines showed a marked slowdown when a certain critical size (both machine- and routine-dependent) is reached (see, for instance, Fig. 4 of their paper).

6.3 The parallel code

The current trend in supercomputer technology is towards achieving raw computational power by assembling a large number of relatively inexpensive nodes, based on mass produced RISC CPUs connected by high-speed data path. Examples are the Origin 2000 by SGI (R10000), the IBM SP/6000 (Power PC) and the Cray T3D (ALPHA). While it is appealing to be able to obtain large theoretical computational speeds at a fraction of the cost of traditional vector based machines, this paradigmatic shift requires a re-examination of the existing codes. A case in point is the spectral Navier-Stokes solver discussed in the previous section. To parallelize it, we begin by noticing that the computationally intensive steps of solving the Helmholtz and Poisson problems amount to solving $i_{max} \times j_{max}$ 1D problems, where i_{max} (j_{max}) is the number of collocation points in the streamwise (spanwise) direction.

The load then can be distributed among p processor. A Single Program Multiple Data (SPMD) approach was adopted. Each process executes essentially the same operations on different portions of the domain, which are private to them. Message passing is used to exchange information between processes, using Message Passing Interface (MPI) library calls.

The domain is sliced along either the z or the x direction, and each processor owns a slice. During the computation of the nonlinear term, the domain is sliced along the z direction (see Fig. 13). When vertical derivatives are needed, a transposition is performed, in which process j sends sub-block i of the domain to process i, and receives in turn the sub-block j from process i. MPI implements this kind of alltoall scatter/gather operation transparently. After the nonlinear term is calculated, the domain is swapped so that each process owns vertical slices, and the Helmholtz and Poisson problems are solved without further swapping. At the end, the solution is swapped back into horizontal slices and the cycle begins again. Incidentally, this approach predates the use of parallel computers, being used for DNS on Cray X-MP to feed the fields into core memory one slice at a time (see, for example, [37]).

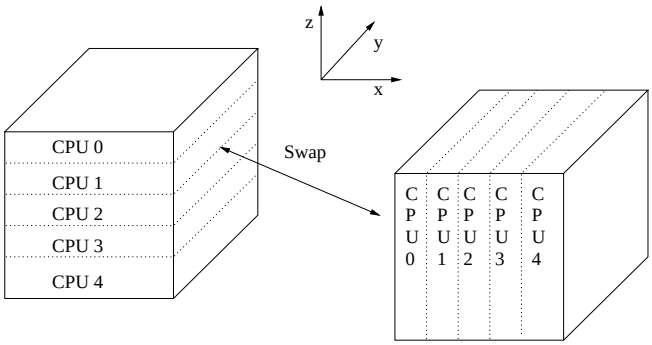

Fig. 13. The domain is split among processes (CPUs) either along the z (left) or the x (right) coordinate. An alltoall scatter/gather is used to go from one configuration to the other.

6.4 Speedup and scalability

The performance of a parallel program is measured by the speedup factor S, defined as the ratio between the execution time T_σ of the serial program and the execution time of the parallel version T_π (see Pacheco [38]). In our approach, the load is evenly balanced between processes (with the negligible exception of I/O, which is handled by one process), so that an equivalent measure is the efficiency E, defined as the ratio between T_σ and the total time consumed by the p processes. In general, for a given machine, $E = E(n, p)$, where n is the size of the problem being solved. In Table 1 we show the efficiency for different values of n and p, with the relative MFLOPS in parenthesis.

The striking result is that it is possible to achieve a super-linear speedup. This is made possible by the fact that the smaller parallel threads use the cache more efficiently than the serial code. For instance, for the grid $128\times128\times96$, the serial code reuses on average a L2 cache line 4.6 times before discarding it;

Size	$p = 2$	$p = 4$	$p = 8$	$p = 16$
32×32×32	.8 (84)			
64×64×64	.93 (81)	.81 (120)		
128×128×96		1.1 (88)	1.3 (168)	
128×192×128			1.1 (168)	.91 (276)

Table 1. Efficiency of spectral parallel Navier Stokes solver and (in parentheses) achieved MFLOP rate.

using 4 processes the value per process increases to 7.3, while with 8 becomes as high as 21.4. The gain is of course offset by the overhead generated by message passing. However, due to the efficient implementation of MPI on the Origin 2000, we found that the time spent swapping data between processes represents less that 10% of the total time, in the worst case.

6.5 Results

The Reynolds number based on channel height and the time-averaged centerline velocity was 7500 for all calculations. Simulations were carried out for several values of the frequency of the driving pressure-gradient, resulting in a Reynolds number, based on the thickness of the laminar oscillating layer, $\delta = (2\nu/\omega)^{1/2}$ and the oscillating component of the velocity, ranging between $Re_\delta = 100$ and 1000. The low Re_δ case was simulated using both a DNS on a $128 \times 128 \times 96$ grid, and an LES using the dynamic eddy-viscosity model [4] on the same domain, discretized using a $32 \times 32 \times 49$ grid. All the other cases were simulated only using the LES approach.

Figure 14 shows the centerline velocity (normalized by the $u_\tau = (\tau_w/\rho)^{1/2}$, where ρ is the fluid density and τ_w is the shear stress at the wall) and τ_w itself. The abscissa is the normalized phase, $\phi = \omega t$. While the centerline velocity is in phase with the imposed pressure gradient, the wall stress is not. At high frequencies a sinusoidal shape is preserved, whereas for low frequencies the distribution of τ_w becomes very asymmetric. This is due to quasi-relaminarization of the flow during the acceleration phase, which is followed by a dramatic instability in the deceleration one. Good agreement with the DNS can be observed.

Figure 15 shows the mean velocity profiles at several phases. Good agreement is again observed between the LES and the DNS for the $Re_\delta = 100$ case. At this frequency a region of reversed flow is present, since the thickness of the oscillating Stokes layer reaches into the buffer layer and the flow reverses near the wall during the decelerating phase (without detachment of the boundary layer). For lower frequencies such reversal is not observed.

Different behaviors of the near-wall region as the frequency is decreased are evident in Fig. 15. A more dramatic illustration of the same phenomena can be seen in Fig. 16, in which contours of the turbulent kinetic energy are shown. At the highest frequency the inner and outer layers appear largely decoupled. A

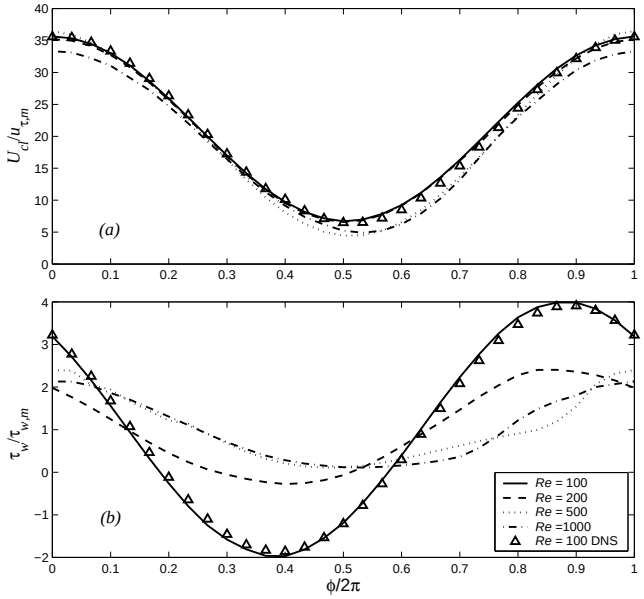

Fig. 14. (a) Centerline velocity and (b) wall stress in the oscillating channel.

thickening of the inner layer can be observed at the end of the deceleration phase ($\phi/2\pi \simeq 0.5$), which, however, does not propagate far into the outer layer: by $z/H \simeq 0.2$ the contours are nearly undisturbed. At lower frequencies, however, the inner layer has the time to adapt to the perturbation introduced by the pressure pulse; at the lowest frequencies in particular the flow can be observed to relaminarize, as indicated by the absence of turbulent kinetic energy. A shift of the more quiescent region of the flow from $\phi/2\pi \simeq 0.8$ towards $\phi/2\pi \simeq 0.5$ can also be observed, which can also be explained based on the increased time that the inner layer has to adapt to the outer-flow perturbation.

The turbulent eddy viscosity, Fig. 17, adjusts to the unsteady perturbation. It is not in phase with the local shear and vanishes as the flow relaminarize during the earlier portion of the accelerating phase. This is in agreement with results from the DNS concerning the evolution of the turbulent kinetic energy production term.

7 Conclusions

Large-eddy simulations have shown the ability to give accurate prediction of the turbulent flow in configurations in which the flow is not in equilibrium, albeit in fairly simple geometric configurations. This type of calculation can now be routinely carried out on desktop workstations, with reasonable throughput times. Parallel computers are required in more complex geometries, in flows in which

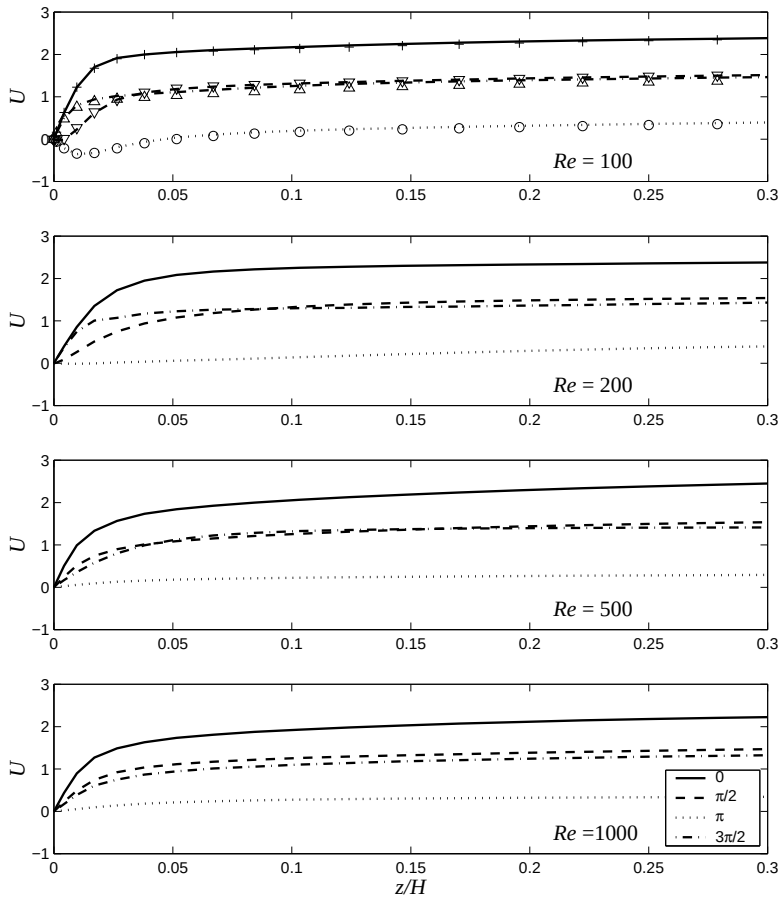

Fig. 15. Velocity profiles in the oscillating channel.

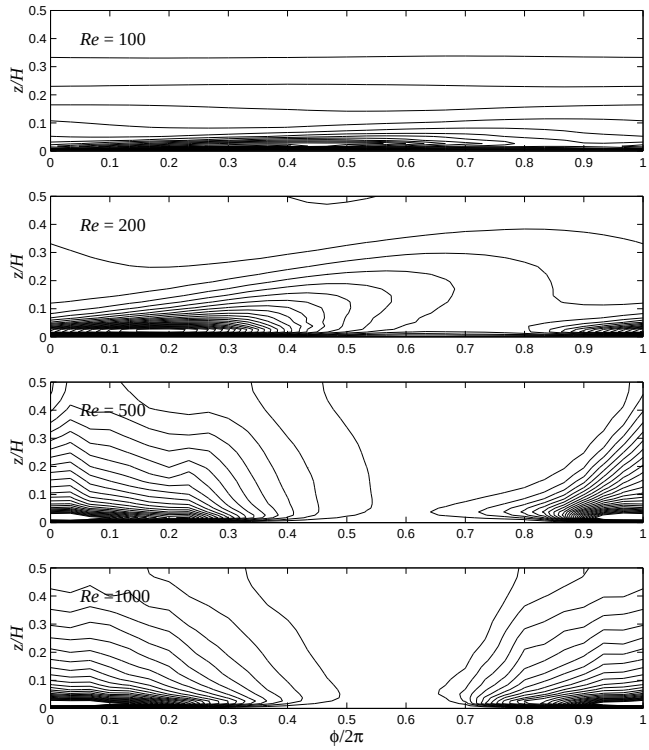

Fig. 16. Contours of the turbulent kinetic energy (normalized by the mean wall stress) in the oscillating channel. 26 equi-spaced contours between 0 and 12.5 are shown

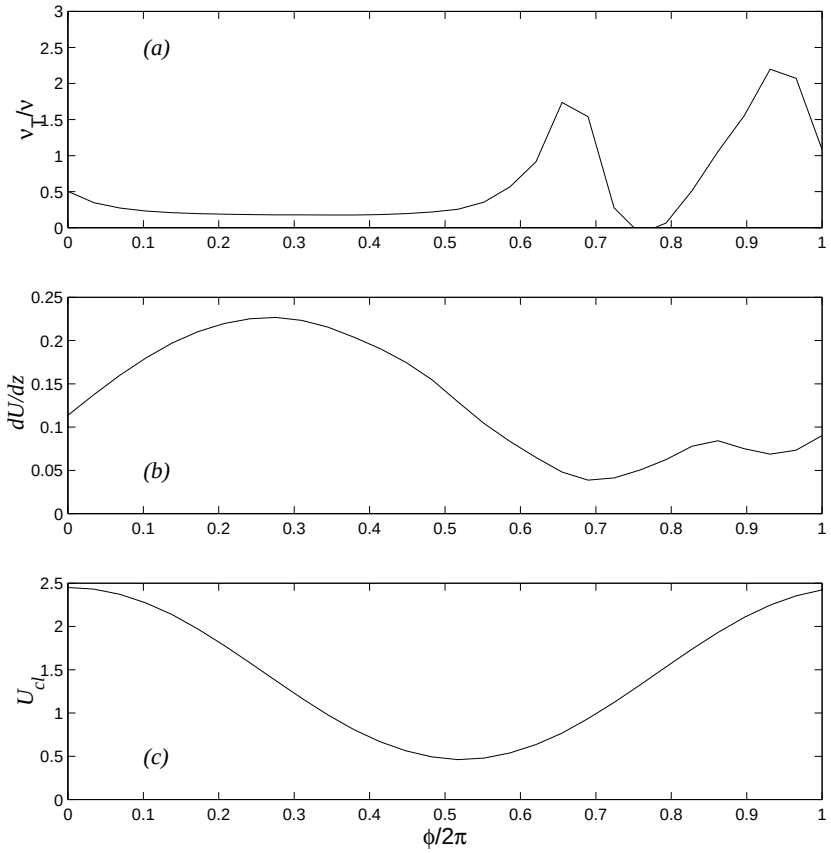

Fig. 17. (a) Phase-averaged eddy viscosity (normalized by the molecular viscosity) at $z/H = 0.0265$. (b) Phase-averaged dU/dz at $z^+ = 15$. (c) Phase-averaged mid-channel velocity. $Re_\delta = 100$.

large computational domains are necessary, and in cases in which long averaging times are required to obtain converged statistics.

The next stage in the development of this technique will involve the use of LES in more complex geometries. Challenges that need to be met to achieve this goal include the development of energy-conserving, high-order schemes in generalized coordinates or on unstructured meshes, and of accurate wall models to simulate the near-wall region without resolving in detail the inner-layer eddies. Combustion models, and SGS models for compressible applications are other areas in which additional research is required to exploit fully the potential of LES. Applications in complex geometries, especially those including combustion, multi-phase flows, or mean-flow unsteadiness, are not likely to be feasible on desktop workstations. Memory-intensive problems will also require parallel machines.

Researchers who use large-eddy simulations are typically end-users of the algorithmic improvements developed by mathematicians and computer scientists. A close collaboration between workers in these fields is, therefore, desirable in order to achieve some progress in the challenging area of turbulence prediction and control.

Acknowledgments

UP and EB acknowledge the support by the NASA Langley Research Center, under Grant No. NAG 1-1828, monitored by Dr. Craig L. Streett. AS acknowledges the support by the National Science Foundation under Grant OCE 99-10883, monitored by Dr. Stephen P. Meacham.

References

1. M. Lesieur and O. Metais. *Ann. Rev. Fluid Mech.* **28**, 45 (1995).
2. U. Piomelli. *Progress Aero. Sci.* **35**, 335 (1999).
3. C. Meneveau and J. Katz. *Annu. Rev. Fluid Mech.* **32**, 1 (2000).
4. M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. *Phys. Fluids A* **3**, 1760 (1991).
5. C. Meneveau, T. S. Lund, and W. H. Cabot. *J. Fluid Mech.* **319**, 353 (1996).
6. G. L. Brown and A. Roshko *J. Fluid Mech.* **64**, 775 (1974).
7. A. Leonard *Adv. Geophys.* **18A**, 237 (1974).
8. J. Smagorinsky. *Mon. Weather Rev.* **91**, 99 (1963).
9. D. K. Lilly. In *Proc. IBM Scientific Computing Symposium on Environmental Sciences.* Yorktown Heights, N.Y., 195 (1967).
10. E. R. Van Driest. *J. Aero. Sci.* **23**, 1007 (1956).
11. U. Piomelli, T. A. Zang, C. G. Speziale, and M. Y. Hussaini. *Phys. Fluids A* **2**, 257 (1990).
12. F. Sarghini, U. Piomelli, and E. Balaras. *Phys. Fluids* **11**, 1607 (1999).
13. J. P. Chollet and M. Lesieur. *J. Atmo. Sci.* **38**, 2747 (1981).
14. J. P. Chollet. In *Turbulent Shears Flow IV*, edited by F. Durst and B. Launder, (Springer-Verlag, Heidelberg), 62 (1984).

15. O. Métais and M. Lesieur. *J. Fluid Mech.* **235**, 157 (1992).
16. F. Ducros, P. Comte, and M. Lesieur. *J. Fluid Mech.* **326**, 1 (1996).
17. Y. Morinishi, T. S. Lund, O. V. Vasilyev, and P. Moin. *J. Comput. Phys.* **143**, 90 (1998).
18. A. J. Chorin. *Math. Comput.* **22**, 745 (1969).
19. J. Kim and P. Moin. *J. Comput. Phys.* **59**, 308 (1985).
20. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral methods in fluid dynamics* (Springer-Verlag, Heidelberg) (1988).
21. Y. Zang, R. L. Street, and J. Koseff. *Phys. Fluids A* **5**, 3186 (1993).
22. Y. Zang, R. L. Street, and J. Koseff. *J. Comput. Phys.* **114**, 18 (1994).
23. Y. Zang, R. L. Street, and J. Koseff. *J. Fluid Mech.* **305**, 47 (1995).
24. P. Beaudan, and P. Moin. *Report No. TF-62*, Dept. Mech. Eng., Stanford University, Stanford, CA 94305 (1994).
25. S. A. Jordan *J. Comput. Phys.* **148**, 322 (1999).
26. A. Kravchenko, P. Moin, and R. D. Moser. "Zonal embedded grids for numerical simulations of wall-bounded turbulent flows." *J. Comput. Phys.* **127**, 412 (1996).
27. D. K. Lilly."A proposed modification of the Germano subgrid-scale closure method." *Phys. Fluids A* **4**, 633 (1992).
28. K. E. Jansen. In *Ann. Res. Briefs–1996*. Center for Turbulence Research, NASA Ames/Stanford Univ., 225 (1996).
29. K. E. Jansen. In *Advances in DNS/LES*, edited by C. Liu and Z. Liu (Greyden Press, Columbus), 117 (1997).
30. D. Knight, G. Zhou, N. Okong'o, and V. Shukla. "Compressible large eddy simulation using unstructured grids." *AIAA Paper 98-0535* (1998).
31. U. Piomelli, E. Balaras, and A. Pascarelli. To appear, *J. of Turbulence*, (2000).
32. T. Sarpkaya. *J. Fluid Mech.*, **253**, 105 (1993).
33. G. Binder, S. Tardu and P. Vezin. ' *Proc. R. Soc. Lond.* A, **451**, 121 (1995).
34. T. A. Zang and M. Y. Hussaini. *Appl. Math. Comput.* **19**, 359 (1986).
35. U. Piomelli. *Phys. Fluids A* **5**, 1484 (1993).
36. M. Frigo and S. G. Johnson. In *ICASSP Conf. Proc.*, **3**, 1381 (1998).
37. R. D. Moser and P. Moin. *NASA TM-85074*.
38. P. S. Pacheco. *Parallel Programming with MPI*, (Morgan Kaufmann, San Francisco) (1997).