COVER SHEET

NOTE: This conversheet is intended for you to list your article title and author(s) name only—this page will not appear on the CD-ROM.

Title: *Large scale simulation of turbulence using a hybrid spectral/finite difference solver* for Proceedings of the 21st International Conference on Parallel Computational Fluid Dynamics (ParCFD 2009)

Authors: Julien Bodart, Laurent Joly, Jean-Bernard Cazalbou

PAPER DEADLINE: ***SEPTEMBER 30, 2009****

PAPER LENGTH: ****16 PAGES (Maximum)****

ABSTRACT

Performing Direct Numerical Simulation (DNS) of turbulence on large-scale systems (offering more than 10^4 cores) has become a challenge in high performance computing. The computer power increase allows now to solve flow problems on large grids (with close to 10^9 nodes). Moreover these large scale simulations can be performed on non-homogeneous turbulent flows. A reasonable amount of time is needed to converge statistics if the large grid size is combined with a large number of cores. To this end we developed a Navier-Stokes solver, dedicated to situations where only one direction is heterogeneous, and particularly suitable for massive parallel architecture. Based on an hybrid approach spectral/finite-difference, we use a volumetric decomposition of the domain to extend the FFTs computation to a large number of cores. Scalability tests using up to 32K cores as well as preliminary results of a full simulation are presented.

INTRODUCTION

Successful attempts of DNS on large-scale systems have been made recently using fully spectral codes with a 2D domain decomposition, referred to as the volumetric decomposition (See for example [7, 13]). Compared to the traditional slabwise decomposition, this method, initiated by Eleftheriou et al. [8], increases the number of usable cores for a problem of size N^3 from N to N^2 . This highly scalable way to perform three dimensional FFTs has been improved and implemented recently in the open source library *P3dfft*, by Pekurovsky [12] at the San Diego Supercomputer Center. An other advantage of the 2D decomposition is the degree of freedom added in the grid management. It gives the opportunity to use unequal number of grid points (N_x, N_y, N_z) without affecting the overall scalability [6]. Thus the grid can be chosen in better agreement with the physics of the flow. We propose to take advantage of the volumetric decomposition in situations where one direction is treated with finite-differences schemes. Such schemes give the possibility (i) to solve non-periodic boundary directions such as those encountered in jet or channel flows for instance (ii) to use nonuniform grids in order to study non-homogeneous turbulence, while keeping the spectral accuracy in two directions.

Université de Toulouse, ISAE
10, avenue Edouard Belin, BP 54032, 31055 Toulouse Cedex, FRANCE

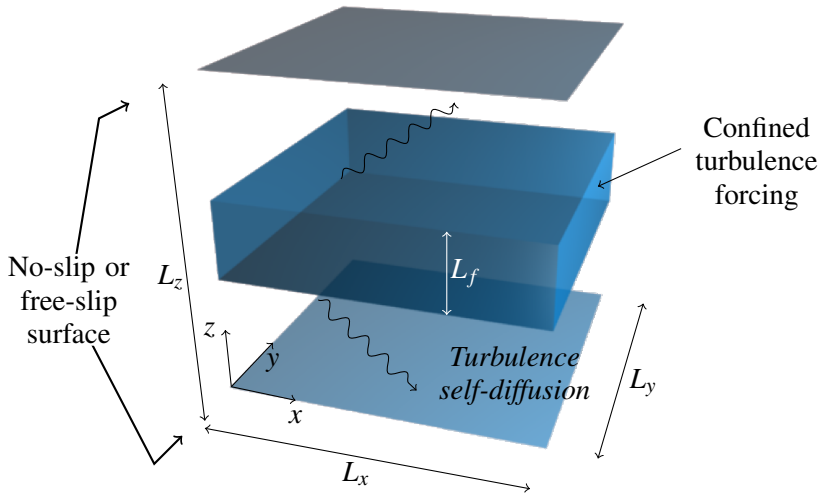

Figure 1. Numerical set up.

With the aim of investigating the interaction between turbulence and a solid wall,we have designed a configuration, where the turbulence self diffuses from a plane source towards a rigid wall as presented Figure 1. The resulting turbulent flow field is homogeneous with respect to x and y, and invariant under rotation about the z axis, which justifies the implementation of such a Navier-Stokes solver. Therefore, the x and y directions are treated using the same number of Fourier nodes N_{xy}, while the z direction use a different number of grid points N_z, and a non-uniform distribution.

The coupling between the possible number of grid points in each directions and the processor distribution is studied and tested on the Blue Gene/P architecture. Scalability of the algorithm is presented using up to 32768 (32K) cores. Finally, preliminary results of a complete simulation using almost 10^9 grid points on 8K cores is presented. From the author's knowledge, it is one of the largest DNS simulation using (i) only two homogeneous directions (ii) a volumetric decomposition in a hybrid spectral/finite difference solver. In this situation where the turbulence is not fully homogeneous, 3 millions CPU hours are required to obtain converged statistics.

FLOW CONFIGURATION UNDER STUDY

In the configuration presented in Figure 1 the turbulence diffuses from a plane source towards a rigid wall. The turbulence production does not rely on the presence of mean shear, but it is instead synthesized inside a bounded layer using a random force. The effects induced by the generation of turbulence by mean shear are therefore not present. The viscous as well as the kinematic blocking effects can be fully isolated in the near wall region as first demonstrated by Campagne et al. [3].

The forcing field is implemented as a source term in the Navier-Stokes equations. It is non-zero only in the central region of the domain (dark-blue on Figure 1), where it appears in the momentum equation as:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i(\mathbf{x},t). \tag{1}$$

It has been designed in the physical space in order to show several properties which are being (i) random in time (ii) divergence-free (iii) localized in space. This kind of forcing demonstrates good abilities to synthesize a plane source of turbulence [2].

The computations are started with a flow at rest. During a transient, the amount of turbulent energy contained in the whole domain grows under the action of the forcing field. A statistically steady state is reached when the power induced by the random force field is statistically balanced by the viscous dissipation. This numerical setup can be seen as an analogue of the oscillating-grid experiments (see for example [5]).

PARALLEL PERFORMANCES OF THE NAVIER-STOKES SOLVER

The Navier-Stokes equations are solved in the hybrid Fourier/physical space. The present solver is pseudo-spectral, based on the use of Fourier modes in the two periodic directions, finite-differences with a sixth-order compact scheme [11] and a staggered arrangement in the wall-normal direction. The parallel implementation is performed in the same way as in a fully spectral solver. In the latter, the parallel efficiency is strongly conditioned by the three dimensional FFT implementation. This algorithm requires two or three different data distributions with the following characteristics: the arrays are distributed over one or two directions in such a way that 3D data with the same array index in the third direction is kept (i) local on a CPU core and (ii) contiguous in memory. Thus, the FFT computation in each direction and/or the finite-difference resolution is performed locally on a single core and does not require any parallel communications. Therefore, the implementation efforts must concentrate on the data transposition from one domain to the other, in order to minimize the cost of the communications involved.

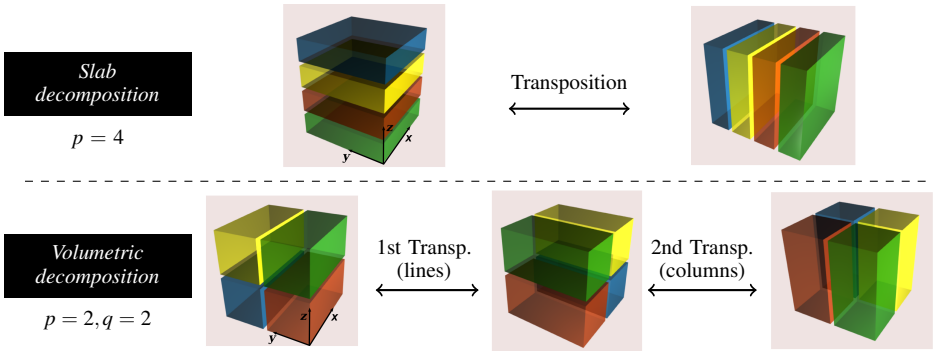

Figure 2. Slab decomposition versus volumetric decomposition.

The slabwise decomposition involves a single transposition and is the most efficient implementation in many cases. With this method however, a problem of size N^3 can be distributed over N cores only, which is a strong limitation on nowadays supercomputers.

Recently, Pekurovsky developed the open-source library *p3dfft* to perform 3D FFT using a volumetric decomposition, meaning that the data are distributed over two instead of one single direction, thus involving $n = p \times q$ cores. This library, written in fortran, calls optimized FFT libraries such as *ESSL* or *FFTW*. Its scalability has been demonstrated on a Blue Gene/L architecture [6]. We use a slightly modified version in our C code to perform 2D FFTs only and solve finite-difference equations

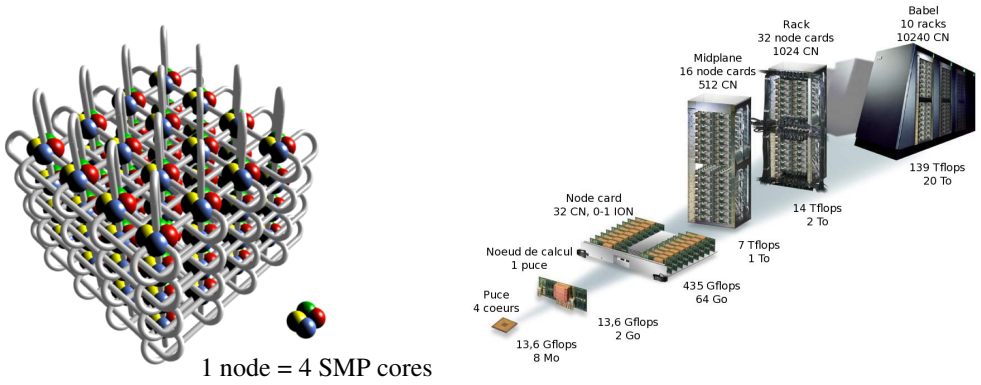

Figure 3. Blue Gene supercomputer and its torus network architecture.

with the optimized Blas/Lapack libraries at the last stage. If $p = q = N$ the number of usable cores increases from N to N^2 for a problem of size N^3. In Figure 4, the data distribution is performed using 4 cores ($p = q = 2$) marked with different colors. We observe that yellow/green and blue/red blocks constitute two independent sets of cores where the parallel communications remain internal during the first transposition. The n MPI processes define a 2D *virtual* $p \times q$ grid where communications are performed independently inside each line/column during the first/second transposition. The Blue Gene/P physical topology is a 3 dimensional torus (X,Y,Z) of quad-cores processors, where each processor can directly communicate with its nearest neighbor in each of the three directions, as presented in Figure 3 (left). In the fully distributed mode (VN), the 4 SMP cores generate a fourth dimension in the topology (X,Y,Z,T), in which the communications times are the fastest. To achieve the best performance during the computation of 2D FFTs, two classes of parameters need to be set to their optimal values:

- the couple (p,q);
- the mapping of a 2D *virtual* process topology on the *physical* cores.

Using the VN mode, selecting the predefined mapping $TXYZ$ and maximizing *the slowest varying index* p in the processor grid gives the best results. In this optimal configuration, processors are numbered using the T direction in the 4D torus as the fastest varying index. Different 2D mappings were tested on a static configuration, with a negligibly smaller restitution time obtained in best cases. In fact, building such a 2D optimized mapping is worthwhile in situations where the parallel communications are mostly of neighbor-to-neighbor type. However, the semi-global communications involved by the transpositions induce many communication types and reach their maximum efficiency in the predefined mapping available on the Blue Gene/P.

LOAD BALANCING

Classically, the data packing chosen in *p3dfft* transform an array with $N_x \times N_y \times N_z$ points in the physical space to a set of ${}^{N_x+2}/_2 \times N_y \times N_z$ Fourier modes in the spectral space (assuming that N_x is even). This data packing introduces a slight load unbalance. Indeed, an optimal distribution of the N_x points over p processors in the physical space will usually result in an unbalanced and non optimal distribution of the ${}^{N_x+2}/_2$ Fourier modes over q processors in the spectral space. Although this is not

	Contiguity in memory	1st direction	2nd direction
1st stage	$(N_z, N_y, N_x/2)$	$N_z = k_1 q$	$N_y = k_2 p$
2nd stage	$(N_z, N_x/2, N_y)$	$N_z = k_1 q$	$N_x/2 = k_3 p$
3rd stage	$(N_y, N_x/2, N_z)$	$N_y = k_4 q$	$N_x/2 = k_3 p$

TABLE I. VOLUMETRIC DECOMPOSITION: COUPLING BETWEEN THE NUMBER OF CORES AND THE NUMBER OF GRID-POINTS.

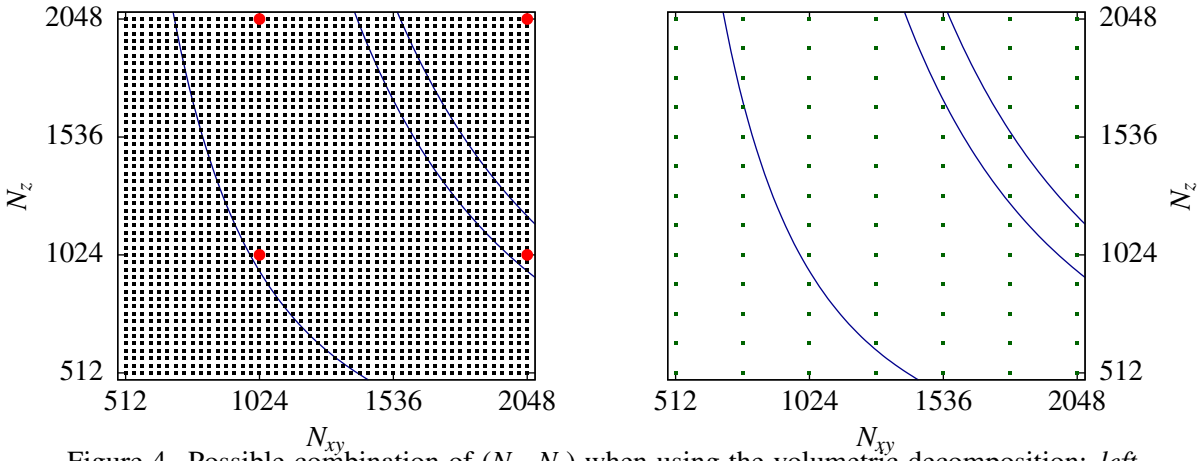

Figure 4. Possible combination of (N_{xy},N_z) when using the volumetric decomposition: *left*- For $n = 1K$ cores. *right* - For $n = 16K$ cores. Red circles indicate the slabwise decomposition ($p = 1$). Blue lines indicate isovalues of the total number of grid-points.

crucial in terms of data exchange between processors, it affects the loop sizes when working locally on a processor at the finite-difference stage. The related extra cost can reach 20% in some cases. To avoid unnecessary operations, we take advantage of the dealiasing method needed to compute the non-linear terms of the momentum equation in the spectral space. The phase-shift method used to this end should be coupled with a spherical truncature [4] which removes in any cases the aforementioned modes $k(N_x+2/2, y, z)$. Consequently, these modes can be safely ignored at the finite-difference stage while preserving the desired accuracy.

GRID DEFINITION

Using the volumetric decomposition, we gain a degree of freedom in the grid generation. It gives the possibility to use different grid-points numbers in different directions, contrasting with the N^3 size imposed by the slabwise decomposition. The grid definition has to fulfill constraints of different kinds in homogeneous/inhomogeneous directions. In the homogeneous directions, it is mandatory to satisfy the spatial decorrelation and resolve the smallest length scale (uniform), whereas in the second case the main issue is to map the grid according to a varying smallest length scale. The new degree of freedom brought by relaxing the constraint $N_z = N_x = N_y$ is particularly useful when the calculation domain should minimize the CPU cost generated by the transient, or when the different directions of the problem are decoupled as in jet flows for instance. In our configuration, the total number of grid points is set as a function of N_{xy} and N_z, corresponding to the dimension x or y, and z. During the different steps of the volumetric decomposition (*i.e.* between the two transpositions), a perfect load

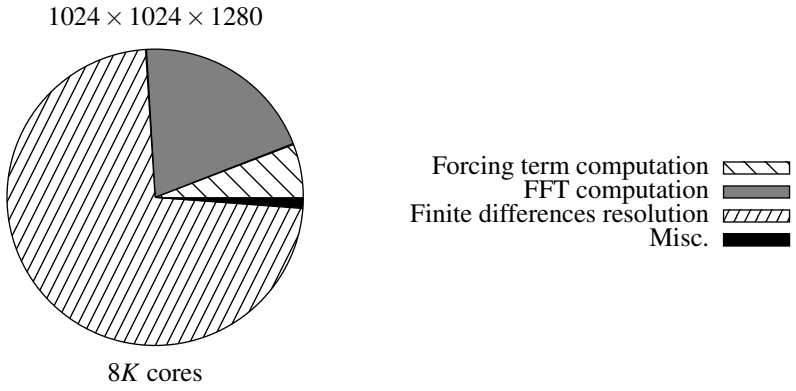

Figure 5. Execution time in several parts of the code.

balancing induces a coupling between the number of grid points and the number of cores. We summarize the resulting constraints in table I using a "C" convention to express the contiguity in memory: (m_1, m_2, m_3) means that m_3 is the fastest varying index, whereas m_1 and m_2 refer to the number of grid-points in the directions splitted using q and p cores respectively. Following these constraints, it follows that an optimal load balancing requires that

$$q \in \mathscr{D}(N_z, N_y) \qquad p \in \mathscr{D}(N_y, N_x/2), \tag{2}$$

where $\mathscr{D}(a,b)$ is the set of common factors of a and b. On Figure 4, we show all the (N_{xy}, N_z) couples satisfying the constraints for both decompositions if the number of cores is set to $n = 1K$ or $n = 16K$. Although this degree of freedom decreases with the number of cores, it considerably extends the meshing possibilities compared to the classic slabwise decomposition, which offers only a few number of possible arrangement (red dots on Figure 4).

PARALLEL PERFORMANCES

In order to evaluate the global scalability of the application, several computations based on different mesh sizes have been ran on a Blue Gene/P computer. The number of grid-points in the different cases, summarized in table II allows to establish two types of diagnostics:

1. Test case A is small enough and requires a modest amount of memory which is compatible with a run using a number of cores confined between 256 and 16K. It offers the opportunity to evaluate the speed-up using a large range of cores number.

2. With the B test cases, we evaluate the performances of the code with respect to the couple (N_{xy}, N_z). Theses computations involve a total number of grid-points (10^9) which is of the same order of magnitude in all cases, and need at least 2K cores to fit in memory.

In order to evaluate precisely the parallel performances of the application, it is mandatory to measure separately the execution times induced by the different parts of the

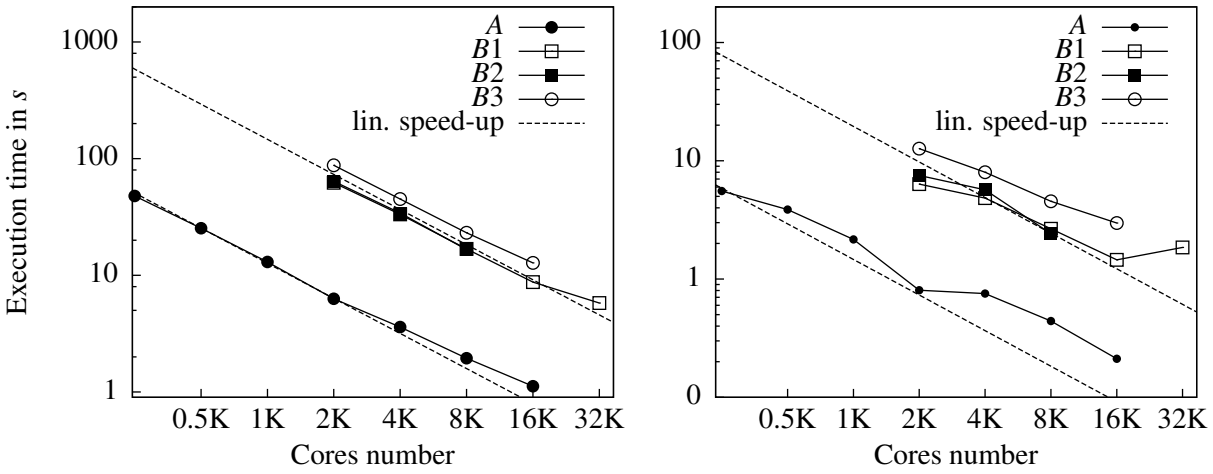

Figure 6. Parallel scaling (1K=1024cores): *left*-Overall scaling, *right*-2D FFT scaling

solver, in such a way to highlight potential bottlenecks in the algorithm. The timing function *gettimeoftheday*, with its low latency compared to the *MPI_Barrier* function, is particularly adapted for such a measure. On Figure 5, we report the time spent in different isolated parts of the code. These results, extracted from the case $B3$, clearly identify two costly parts of the code: the FFT computation and the resolution of linear systems brought on by the finite-difference approach used in the third direction. Thanks to the transpositions, the linear systems are solved locally on each core. Therefore, the most expensive step does not involve any parallel communications. Conversely, a complete FFT computation which include several transpositions, requires data exchange between processors and should limit the scalability of the application. We therefore present two sets of results on Figure 6, one being related to the overall computation while the other being restricted to the FFT computations and data transpositions. As expected, the second part does not scale perfectly (right), while demonstrating very good performances with up to 16K cores. As a consequence, the whole application scales almost perfectly (left), as most of the work is done locally on each core. Furthermore, modifying the values of the couple (N_{xy}, N_z) does not appear to significantly affect the performance, as it does not change the slope of the curves between the B cases. The hybrid spectral/finite-difference formulation coupled with a volumetric decomposition is therefore perfectly adapted to a massively parallel architecture.

Test case	A	$B1$	$B2$	$B3$
N_{xy}	512	1024	896	1024
N_z	512	1024	1280	1280

TABLE II. TEST CASES PRESENTED TO ESTIMATE THE SOLVER OVERALL SCALABILITY.

APPLICATION USING A $896 \times 896 \times 1024$ GRID.

The above implementation has allowed to run a direct numerical simulation of the flow on a $896 \times 896 \times 1024$ grid, and to perform a statistical analysis of the con-

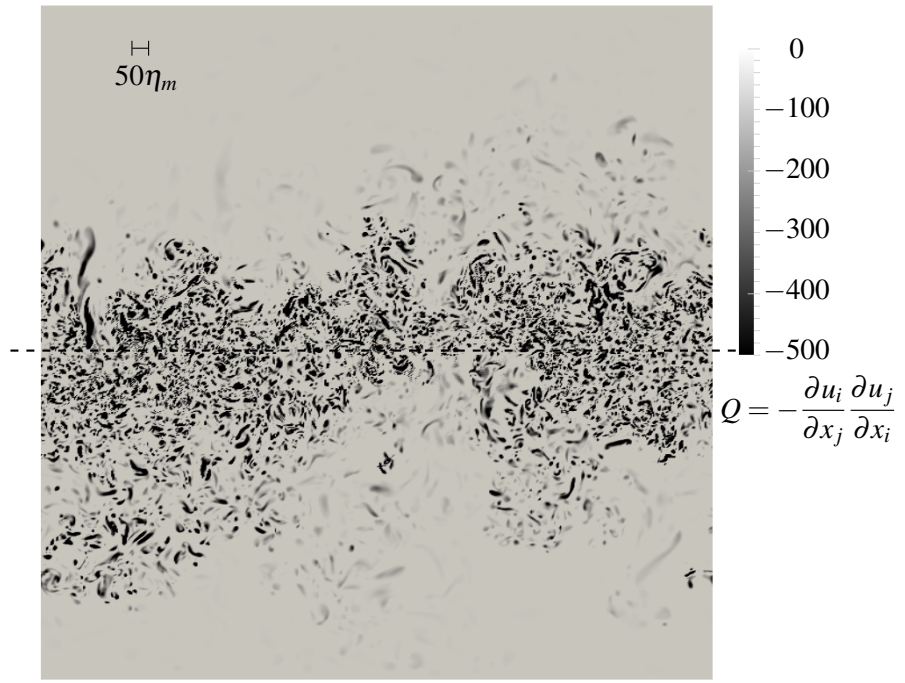

Figure 7. Negative values of the Q criterion for coherent structure identification in a plane cut, normal to the wall. η_m express the value of the Kolmogorov length scale in the midplane (dashed line).

figuration presented in Figure 1. We present here the partial results to illustrate the developed algorithm. More in depth analysis of the configuration can be found in [3] or [1].

An overview of the configuration is presented in Figure 7. The turbulence agitation is qualitatively described using the Q criterion [9] applied on an instantaneous flow field. Negative values should be linked to coherent rotating structures. In this plane normal to the walls we identify (i) the central region where the turbulence is produced (ii) the pure diffusion region on both sides. In the second region, the intensity of the agitation decreases, while the characteristic length scale of the structures increases. According to the Kolmogorov's similarity hypothesis, the turbulence energy density spectrum should exhibit a $-5/3$ slope, which can be observed if the length scale separation between the large (energy containing) and the small (dissipative) length scales is sufficiently high. The latter condition is obtained if the turbulent Reynolds number, $Re_t = K^2/\nu\varepsilon$[1] reaches a sufficiently large value, which is directly linked to the turbulent agitation the grid size can support. Only very large direct simulations allow to observe this "inertial subrange". The evaluation of the Kolmogorov's constant is even more difficult [10]. In our study, only two directions are homogeneous, which leads to introduce a 2D energy-density spectrum instead of the traditional 3D spectrum used in the context of fully homogeneous turbulence. The 2D FFTs performed in each homogeneous plane (xy) gives access to a 2D spectrum E_{2D} which satisfies:

[1] K and ε being respectively the averaged turbulent kinetic energy and the averaged dissipation rate

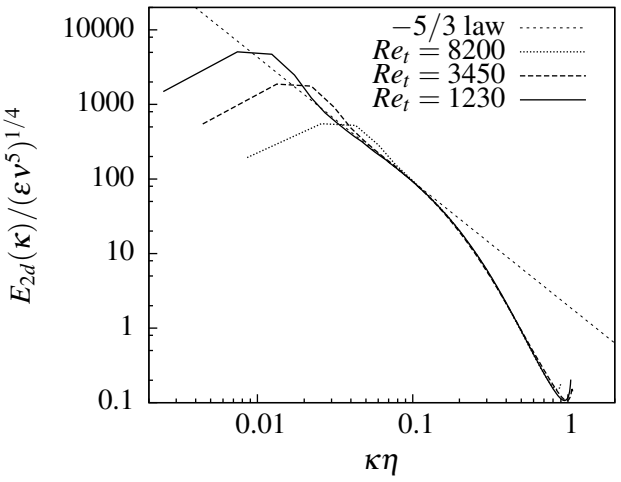

Figure 8. Energy spectrum in the midplane for three different Reynolds numbers.

$$\iint_0^\infty E_{2D}(\kappa,z)d\kappa = K(z), \tag{3}$$

where κ refers to the modulus of the two-dimensional wavenumber (k_1,k_2). The shape of the Kolmogorov spectrum being established from dimensional considerations only, the $-5/3$ slope still holds with E_{2D}. In Figure 8, we restrict our observations to the midplane, where the turbulent agitation reaches a maximum. We also consider three different simulations characterized by different values of their turbulent Reynolds numbers in the midplane. The higher value corresponds to the grid size discussed above, while the other computations require much less grid points. The inertial subrange is remarkably obtained at the higher Reynolds number value, even if the configuration is anisotropic at the largest scales. Thus, the forced turbulence exhibit realistic physical properties aside the bump at the largest scales which reflects the action of the forcing term at these scales. Besides, we demonstrate a perfectly similar behavior at the smallest scales with a Reynolds number value increased by a factor of ten.

CONCLUSION

The volumetric decomposition is shown to be particularly adapted to a mixed hybrid/finite-difference solver, as demonstrated by the large-scale scalability test presented here. Furthermore it provides an additional degree of freedom in the grid generation, which is particularly adapted when studying non-homogeneous turbulent flow fields. The proposed implementation has been used to perform a large-scale direct simulation which involve close to 10^9 grid points and 3 millions CPU Hours. The turbulent Reynolds number in this simulation is of the order of 10^4, allowing the development of a convincing inertial subrange in the energy-density spectrum, which makes the physical analysis more meaningful with respect to practical applications.

ACKNOWLEDGMENTS

Preliminary computations were carried out under the HPC-EUROPA++ project (project number: 1187), with the support of the European Community - Research Infrastructure Action of the FP7 "Coordination and support action" Program. Final computations have been carried out on the Blue Gene/P supercomputer at IDRIS (Paris).

REFERENCES

1. J. Bodart, J.-B. Cazalbou, and L. Joly. Direct numerical simulation of unsheared turbulence diffusing towards a free-slip or no-slip surface. In *Turbulence and Shear Flow Phenomena 6*, June 2009.
2. J. Bodart, L. Joly, and J.-B. Cazalbou. Local large scale forcing of unsheared turbulence. In *Direct and Large Eddy Simulation 7*, September 2008.
3. G. Campagne, J.-B. Cazalbou, L. Joly, and P. Chassaing. The structure of a statistically steady turbulent boundary layer near a free-slip surface. *Physics of Fluids*, 21(6), June 2009.
4. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral methods in Fluid Dynamics*. Springer-Verlag, 1988.
5. I. P. D. De Silva and H. J. S. Fernando. Oscillating grids as a source of nearly isotropic turbulence. *Physics of Fluids*, 6(7):2455–2464, 1994.
6. D. A. Donzis, P. K. Yeung, and D. Pekurovsky. Turbulence simulations on o(10^4) processors. June 2008.
7. D. A. Donzis, P. K. Yeung, and K. R. Sreenivasan. Dissipation and enstrophy in isotropic turbulence: Resolution effects and scaling in direct numerical simulations. *Physics of Fluids*, 20(4), 2008.
8. M. Eleftheriou, J. E. Moreira, B. G. Fitch, and R. S. Germain. A Volumetric FFT for BlueGene/L. pages 194–203. 2003.
9. J. C. R. Hunt, A. A. Wray, and P. Moin. Eddies, streams, and convergence zones in turbulent flows. In *Proceedings of the 1988 Summer Program*, pages 193–208, December 1988.
10. T. Ishihara, T. Gotoh, and Y. Kaneda. Study of high–reynolds number isotropic turbulence by direct numerical simulation. *Annual Review of Fluid Mechanics*, 41(1):165–180, 2009.
11. S. K. Lele. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103(1):16–42, November 1992.
12. D. Pekurovsky. P3dfft open source library.
13. J. Schumacher and M. Putz. Turbulence in laterally extended systems. In *Parallel Computing, Architectures, Algorithms and Applications*, volume 38, pages 585–592. John Von Neumann Institute for Computing, 2007.