Turbulence Simulations on $O(10^4)$ Processors

D.A. Donzis[1,2], P.K. Yeung[2] and D. Pekurovsky[3,4]

[1]Institute for Physical Science and Technology, University of Maryland, College Park, MD 20742

[2]School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332

[3]San Diego Supercomputer Center, La Jolla, CA 92093

[4]University of California San Diego, La Jolla, CA 92093

1 Introduction

Turbulence is the most common state of fluid motion in nature and engineering, and a very difficult subject in the physical sciences (see, e.g., [1],[2]). Advances in understanding turbulence and the ability to model its effects are important in diverse applications such as aerospace vehicle design, combustion processes, and environmental quality. Turbulent flows are characterized by disorderly nonlinear fluctuations in time and three-dimensional space, which often limits the degree of detail that can be measured reliably in laboratory experiments. A very powerful research tool is Direct Numerical Simulation (DNS, see., e.g, [3]), where fluctuations at all scales are computed according to exact equations of mass and momentum conservation, and many important quantities can be extracted. The main difficulty in DNS is the fact that the range of scales, and hence the computational demands in both speed and memory, increases strongly with the Reynolds number, which is usually high in applications. Numerical simulation of high Reynolds number turbulence is known as a grand challenge in high-performance computing, with tremendous opportunities for scientific progress anticipated in the future [4].

Although turbulent flows in practice often occur in complex geometries, classical hypotheses of scale similarity [5] suggest that key aspects of the flow physics, especially at the small scales, can be studied effectively in the simpler configuration of homogeneous isotropic turbulence, which has no spatial dependence or preferential orientation in its statistical properties. Computationally this is also highly convenient because a discrete Fourier representation on a periodic domain becomes applicable, thus allowing the use of highly efficient and accurate pseudo-spectral methods for the numerical solution. These solution methods allow flow variables at different modes in transformed space to be formally decoupled, and are thus are well adapted for distributed-memory processing on computer platforms with a large number of parallel processing elements. However, especially with present trends in high-end computing towards extreme parallelism, it is clear that an efficient and highly scalable domain decomposition scheme is vital. In particular, despite previous successes on TeraGrid machines using $O(10^3)$ processors, the best use of machines with (10^4) processors, beginning with the first NSF Track 2 system (Ranger at TACC), is expected to require new strategies.

This paper has two main purposes. The first is to describe the formulation of a new domain decomposition scheme and report on the scalability characteristics of a new DNS code based on this decomposition on multiple platforms, with emphasis on performance achieved to-date on Ranger. The domain decomposition employed is two-dimensional (2D), such that an N^3 problem can be simulated using up to N^2 processors, which would not be possible with a simpler 1D decomposition. This code has been run on Ranger at 4096^3 resolution using 16K cores, with 87% strong scaling for a quadrupling of core count from 4K to 16K. Testing at large core counts has also been performed on IBM BG/L and CRAY XT4's at other major supercomputing sites, with 98% strong scaling observed between 16K and 32K cores on the former.

Our second main purpose here is to report on early results from two series of production runs at 4096^3 resolution, which are the first published in the US at this scale. The first of these is directed at developing a statistically stationary velocity field at higher Reynolds number than before, as a key prerequisite for high-Reynolds-number studies of turbulent mixing of passive scalars representing passive contaminants transported in the flow [6], or of dispersion in a Lagrangian frame following the trajectories of infinitesimal fluid elements [7]. In this case, computing power is used to increase the range of scales present in the simulation. The other simulation is, in contrast, aimed at better resolving the smallest scales better, which, as we have shown in a recent publication [8], is essential for fully capturing the characteristics of fine-scale

intermittency [9]. The latter is important, for example, in the nature of intense but localized fluctuations of energy dissipation rate, which have a strong effect on local extinction and reignition phenomena in turbulent combustion.

The rest of the paper is organized as follows. In the next section we describe the numerical scheme and the domain decomposition of the new code, as well as current performance data (subject to future optimizations) on Ranger and other TeraGrid machines. In Sec. 3 we discuss progress made from production simulations at 4096^3 designed for the purposes indicated above. Finally in Sec. 4 we present the conclusions.

2 Computational Algorithms

2.1 Numerical Method and Domain Decomposition

We compute instantaneous velocity and pressure fields according to basic principles of conservation of mass and momentum. The (Navier-Stokes) equations solved are here written in the form

$$\partial \mathbf{u}/\partial t + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla(p/\rho) + \nu \nabla^2 \mathbf{u} \,, \qquad (1)$$

with constant density ρ (which requires $\nabla \cdot \mathbf{u} = 0$) and viscosity ν. The advantages of accuracy and efficiency of pseudo-spectral methods for partial differential equations of this type are well known [10]. Nonlinear terms are computed in physical space, with aliasing errors controlled by truncation and phase-shifting techniques [11]. In Fourier space Eq. 1 is transformed to ordinary differential equations which are advanced in time by an explicit second-order Runge-Kutta method, with a Courant number constraint for numerical stability which typically makes the time step Δt very small compared to physically relevant time scales. The solution domain is a periodic cube, and most of the computational cost is in 3-D Fast Fourier Transforms (FFTs).

The key to using more parallel CPUs to tackle bigger problems is in the domain decomposition scheme used to divide an N^3 grid into M (usually) equal-sized subdomains. The simplest scheme is a 1D decomposition, where these sub-domains would be slabs of N/M planes in say, $x-y$ directions where transforms in x and y can be taken on data residing entirely locally in each processor's memory. To complete the transform in z, the data are transposed into $x-z$ planes. These transposes (and their inverses, for the inverse transforms) are performed using MPI_ALLTOALL collective communication among all the processors. This method has worked well for grid resolution up to 2048^3 (e.g. [12, 13, 14]), but is obviously restricted to $M \leq N$ (with M being a integer factor of N).

As the problem size increases, on a given machine the restriction $M \leq N$ above will eventually lead to either more memory required than available per processor core, or clock times longer than desirable for production purposes. Accordingly, and given the fact that machines (such as Ranger) at the top end of the spectrum now offer tens of thousands of processing elements, we have implemented a 2D decomposition, which divides the data instead into "pencils", each of size $N \times (N/M_1) \times (N/M_2)$ where $M_1 \times M_2 = M$ define the processor grid geometry. An example of a 4×4 scheme for 16 cores is illustrated in Fig. 1. Two communication calls are needed per transform, e.g., to switch the data from pencils aligned in x, to those aligned in y, and finally to those aligned in z. However these transposes only involve groups of M_1 tasks within the same *row* communicator (e.g. ({0,4,8,12} in the figure), or of M_2 tasks within the same *column* communicator (such as {0,1,2,3} here). For generality, these calls are implemented using MPI_ALLTOALLV.

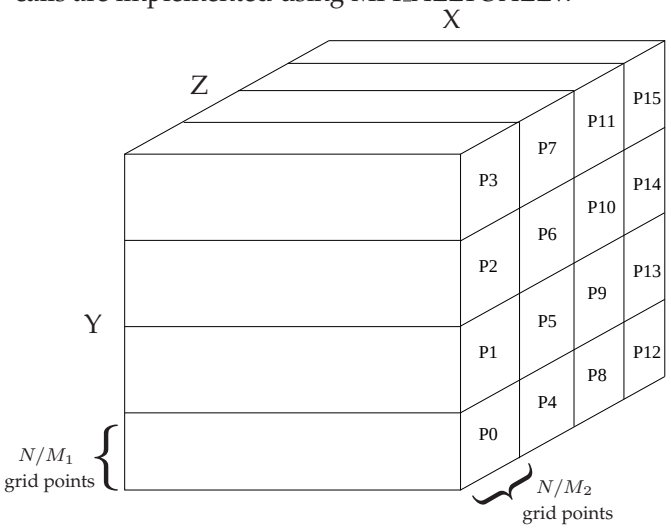

Figure 1: Two-dimensional domain decomposition with $M_1 = M_2 = 4$ ($M = 16$).

It is clear that, for an N^3 problem, the 2D, or pencils, decomposition scheme has the important advantage of allowing the use of up to N^2 processors (in order words, well over 1 million) instead of N. This is also a key prerequisite for computing in the future Petascale era, which is expected to feature core counts in the range $O(10^5)$ or even higher. On the other hand, the actual sustained performance is now subject to a larger number of independent factors, in particular, not just the problem size (N^3) and degree of parallelism (M), but also on the shape of the processor grid (M_1 versus M_2), with details depending on the machine. Since maximum core counts available may vary, we have written our codes to allow also for cases where M is not a power of 2, as long as M_1 is still chosen as an integer factor of N. In such cases some processors will carry a slightly larger workload, but de-

parture from perfect load balance is still small.

Although our current focus is on the simplified problem of homogeneous isotropic turbulence on a cubic domain, our implementation also allows for cases with unequal numbers of of grid points (N_x, N_y, N_z) in each direction, which would be appropriate in more general problems with different length scales in different directions. Furthermore, since 3D FFTs are relevant to many other fields of science, a FFT kernel with all of these features, as developed by one of the authors (DP) has been made publicly available as an open-source library on the web [15].

2.2 Code performance

Our DNS code with 2D decomposition has been tested and benchmarked extensively on a number of leading platforms at several nationally-ranked supercomputing sites. Subject to multicore optimizations likely to become increasingly important in the future, the code is highly portable with the main machine-dependent element being only FFT library calls taken one direction at a time, using the latest version of FFTW or (on IBM machines) the IBM ESSL library. Since FFTs account for most of the computational time, we suppose that perfect scalability would be indicated if the CPU (in wall clock) per time step scales with

$$N^3 \log_2 N/M \ , \qquad (2)$$

by a proportionality factor that depends on the number of operations per grid point and the speed of the machine hardware. As is widely understood, strong scaling implies CPU/step proportional to $1/M$ (with N fixed), whereas weak scaling would be indicated by CPU proportional to $\log_2 N$ if the ratio N^3/M is kept constant.

A summary of our benchmarking data is given in Fig. 2, where we show CPU time per step (in seconds, of wall time) as a function of number of processors (M) on several different platforms, with single-precision arithmetic for grid resolutions 2048^3, 4096^3 and 8192^3. Overall, the code shows high scalability up to very large number of processors. For example, on IBM BG/L's we observe weak scaling of 85% from $(N, M) = (2048, 4096)$ to $(4096, 32768)$, and strong scaling as high as 98% for fixed $N = 4096$ from $M = 16384$ (39 secs.) to $M = 32768$ (19.8 secs.). However, interpretations of scalability based on these data are subject to effects of the shape of the 2D processor grid, which affects both communication and (via use of memory cache) computation times. On machines with 3D-torus network topologies (including IBM BG's and Cray XT4) we observe a systematic trend for better timings when M_1 is made small compared to M_2 (for given N and M). The timings presented in this paper (Fig. 2 and elsewhere) are those obtained using the best processor grid geometry in each case which would obviously be employed for production runs. The data here also includes the case (on IBM BG/L) $(N, M) = (2048, 6144)$, which is an example of processor counts not being perfect power of two and is seen to deviate only slightly from the main data trends.

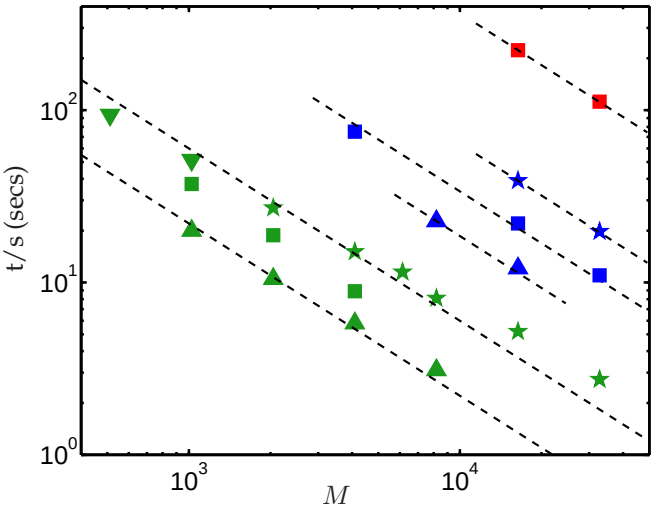

Figure 2: DNS timings obtained for 2048^3 (green), 4096^3 (blue) and 8192^3 (red) grids on Ranger at TACC (squares), Franklin (Cray XT4) at NERSC (upright triangles), IBM Blue Gene/L at SDSC and IBM Watson Research Center (stars), and DataStar (IBM SP4) at SDSC (down triangles). Dashed lines of slope -1 represent limit of perfect scalability, for comparison.

We devote more attention here to performance on Ranger at TACC, which since its deployment has been the largest TeraGrid platform (and one of the world's most powerful supercomputers). Since the machine has as many as 62,976 cores (3936 nodes of 16 cores each) achieving its full potential for our purposes requires successful scaling to large core counts, in the order of 16384 or 32768. As one of the Early User groups on the machine we have been active since December 2007 in working on a better understanding of various factors affecting code performance, including different implementations of the MPI library, the effects of processor grid geometry, and the multi-core nature of processors on Ranger. For example, since Ranger has 16-core nodes, there are significant advantages in choosing M_1 to be 16 or an integer factor thereof, because this would allow transposes involving *row* communicators to be taken among cores which share a local network link and thus should be faster than communication between cores across different nodes. For 4096^3 simulations on 16384 cores we have tested processor grid configurations such as 4×16384, 8×2048 and 16×1024, and determined the first of these to be consistently the best. A possible reason may be related to the fact that each 16-core node is made up of 4 sockets (of 4

cores each). With $M_1 = 4$ the row communication occurs within the socket, so no links to other sockets' memory banks are used. Given that Ranger is (unlike IBM BG and Cray XT4 of which multiple installations are in operation) still relatively new and unique, it is not surprising that some of these benchmarking data await more rigorous explanation in the future.

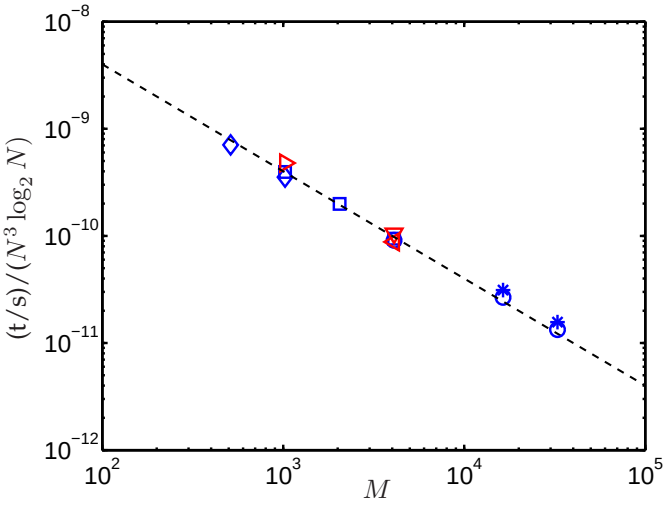

Figure 3: Parallel scaling of the DNS code on Ranger. Time per step (t/s) normalized according to Eq. 2 on Ranger. Blue symbols for N^3; red symbols $N \times 4N \times 4N$ grids. Diamonds: 1024^3; squares: 2048^3; circles: 4096^3; asterisks: 8192^3; right triangle: 256×1024^2; left triangle: 512×2048^2; down triangle: 1024×4096^2. Results at $M = 4096$ overlap and are partly hidden. Dashed line with slope -1 for comparison.

In Fig. 3 we show normalized timings on Ranger only, for several different problem sizes, in terms of CPU/step divided by $N^3 \log_2 N$ which according to Eq. 1 would behave as a straight line of slope -1 if perfect scalability is achieved. For 2048^3 simulations our data show almost 100% strong scaling as the number of processors is quadrupled from 1024 to 4096. For the largest problem sizes, 4096^3 and 8192^3, we also observe virtually perfect strong scaling when the number of cores is increased from 16384 to 32768. Scaling close to 95% is also found as both N and M grow from $(N, M) = (2048, 2048)$ to $(4096, 16384)$ (18.8 and 22 secs. respectively) and 93% from $(2048, 1024)$ to $(4096, 16384)$. The ratio in CPU time between Ranger (75 secs) and Lonestar at TACC (71 secs) is also more more favorable than what their per-core theoretical peak ratings suggest (a ratio of 10.6 Gflop/s on Lonestar to 8.0 Gflop/s on Ranger).

In Fig. 3 we have included data for non-cubic domains which are relevant to more complex flows as stated at the end of Sec. 2.1. Benchmarks were obtained for grids of shape $N \times 4N \times 4N$, for which we define an "equivalent cubic" N_e^3 grid where $N_e = (N \times 4N \times 4N)^{1/3}$ and use Eq. 1 where the N within is taken as N_e. Data for $N = 256$, 512 and 1024 interpreted in this way are also shown in Fig. 3 and are seen to indicate high scalability as well.

Since memory on Ranger (at 2 GB/core) is sufficient for 4096^3 on 4K cores we have also made a comparison between timings in this case obtained with 1D and 2D decompositions. The former is found to be 103 secs/step, which is 36% more than the 76 secs/step based on the 2D decomposition. This again emphasizes that a 2D decomposition is a necessity for large problems.

In addition to computation and communication, large-volume I/O associated with checkpointing N^3 datasets can, depending on I/O node availability and filesystem performance, also be very significant. Currently each MPI task writes its own file, each of size $3N^3/M$ words considering three velocity components. For 4096^3 runs on 4K cores the time taken to read or write 768 TB of data (in single-precision) on Ranger (which has a Lustre filesystem) is about 25 secs. This is not a concern, and in fact implies an impressive I/O bandwidth of 30 GB/sec.

The very good scaling observed on a wide variety of platforms overall shows that our DNS code based on 2D domain decomposition is suitable for very large scale simulations. Although performance at 16384 cores and beyond is not yet highly reproducible on Ranger, we were able to run a preliminary test at 32768 cores with 13 secs/step for a 4096^3 grid which shows strong scaling of about 85% from 22 secs/step at 16384 cores. However, further tests should be conducted especially to determine the optimum process grid for this problem size and processor count. We also expect performance to improve further in the next months as the system progresses towards maturity, and as detailed execution profiling analyses become available.

3 Science Results

We present several preliminary results from the two 4096^3 simulations in progress on Ranger as stated in Sec. 1: we refer to these as Simulation I, which gives Taylor-scale Reynolds number $R_\lambda = O(840)$ with grid spacing $\Delta x/\eta \approx 2$, and Simulation II which is at $R_\lambda = O(630)$ with $\Delta x/\eta \approx 1$. Both simulations can be compared with work [8] up to $R_\lambda \approx 650$ with $\Delta x/\eta \approx 2$. Although final results require sampling over longer periods of time, the early results are promising.

Classical "K41" Kolmogorov scaling [5] states that at *sufficiently high* Reynolds numbers, there is a range of length scales —the *inertial range*—much smaller than the energy-containing scales (L) and much larger than the dissipative scales (η, named after Kolmogorov), where

the dynamics is independent of both the large scale geometry and viscous effects at the small scales. The energy spectrum, which expresses the distribution of turbulence kinetic energy (TKE) among different scale sizes ($1/k$, where k is wavenumber) is then given by the so-called "five-thirds" law, i.e.,

$$E(k) = C\langle\epsilon\rangle^{2/3}k^{-5/3}, \qquad (3)$$

where C is known as the Kolmogorov constant, and $\langle\epsilon\rangle$ is the mean dissipation rate of TKE per unit mass. In Fig. 4 we show compensated spectra in which (according to Eq. 3) the inertial range would be seen as a plateau at intermediate scales. Clearly, the inertial range widens with increasing Reynolds number (R_λ). The 4096^3 simulation gives well over one decade of inertial range, which will allow us to study the details of mixing and dispersion at high Reynolds number much more definitively than before. Towards the high wavenumber end the spectrum does show a turnup which is a consequence of residual aliasing errors but is not sufficiently large to affect the conclusions we draw in this paper.

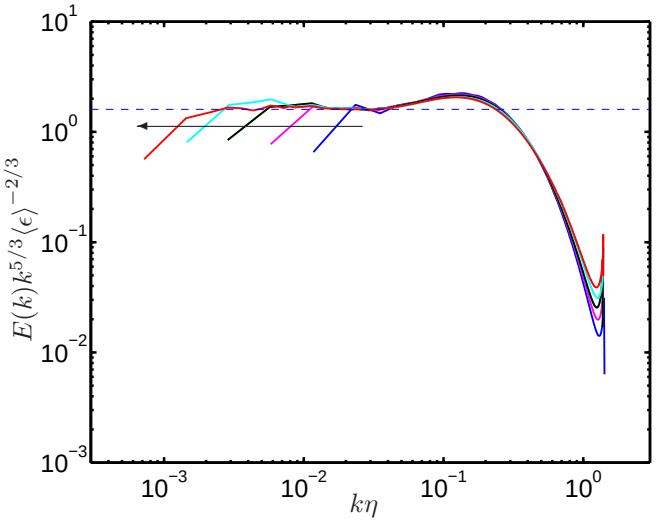

Figure 4: Energy spectra in Kolmogorov variables at R_λ≈140 (256^3, blue), 240 (512^3, magenta), 400 (1024^3, black), 650 (2048^3, cyan) and 840 (4096^3, red); The last case shown is from new Simulation I. Black arrow indicates trend of increasing R_λ. Dashed line at 1.62, as suggested by experimental data.

The spatial structure of turbulence is well characterized by the statistics of velocity increments $\Delta_r u$ between two points in space, at a distance r apart with r in different scale ranges. The inertial range result For the n-th order moment of $\Delta_r u$, called structure function of order n, the inertial-range result (for $\eta \ll r \ll L$) is of the form

$$\langle(\Delta_r u)^n\rangle \equiv \langle[u(x+r) - u(x)]^n\rangle \sim (\langle\epsilon\rangle r)^{n/3}\,. \qquad (4)$$

Structure functions are also classified as longitudinal or transverse, depending on whether velocity components along or perpendicular to the separation vector between the two observation points are considered. The case of the third-order ($n = 3$) longitudinal structure function is of special interest since an exact value (-4/5) is available for the proportionality coefficient, thus providing a convenient test for the attainment of inertial-range scaling.

In Fig. 5 we show structure functions of odd order with $n = 3, 5, 7, 9, 11$ normalized by $(\langle\epsilon\rangle r)^{n/3}$ in each case. For small r there is some apparent irregularities at higher orders, which is an indication of statistical variability that we expect to diminish when longer-time averages become available as the simulation is extended. Nevertheless, for $n = 3$ we see, again, Kolmogorov scaling for about a decade with values close to $4/5$ (horizontal dashed line). However, for high-order structure functions departures from K41 are apparent as they do not show a plateau at inertial range scales. Instead, close fits are obtained with sloping dashed lines representing scalings of the form $\langle(\Delta_r u)^n\rangle \sim r^{\zeta_n}$ (where the ζ_n's are called scaling exponents) with $\zeta_n < n/3$. The departures of these observed scalings from Eq. 4 are of central interest in the development of various theories, including those based on multifractal concepts, to account for intermittency corrections to classical K41 phenomenology, especially for higher-order statistics and at higher Reynolds numbers.

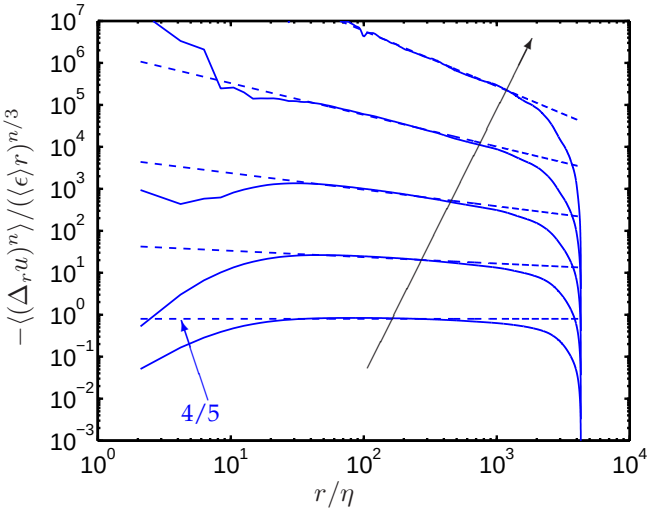

Figure 5: Scaling of odd-order structure functions from new Simulation I, at R_λ≈840 (4096^3) of orders n = 3, 5, 7, 9 and 11. Black arrow indicates trend of increasing n.

The phenomenon of intermittency can be described as the tendency of a quantity to exhibit large but localized fluctuations in space and time especially at high Reynolds numbers. For example it is useful to look at the form of the probability density functions (PDFs) of fluctuations of the energy dissipation rate (ϵ, proportional to strain-rate squared) and enstrophy (Ω, proportional to

vorticity squared). These variables represent different aspects of the small-scale motions and have the same mean value in homogeneous turbulence, but in all available datasets at finite Reynolds number their other statistics are known to differ. We have recently shown [8] that as the Reynolds number increases to large values the PDFs of ϵ and Ω remain different for fluctuations about 100 times the mean, but, remarkably behave in almost the same way in the range of extreme fluctuations at 1000 times the mean and higher. Both of the new simulations confirm the validity of this observation, as seen in the figure. However, caution is needed because a grid spacing $\Delta x \approx \eta$ is still not small enough for higher-order statistics, and longer simulations are needed to produce reasonably-converged statistics, especially for extreme fluctuations whose samples are inherently rare.

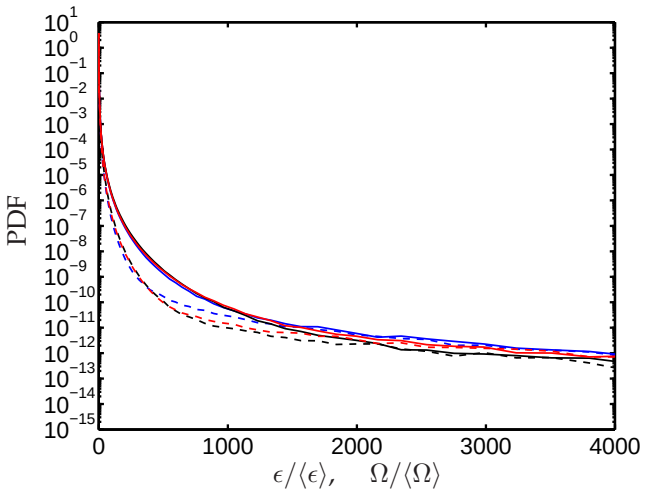

Figure 6: Probability density function for energy dissipation rate ϵ (dashed line) and enstrophy Ω (solid line) at $(R_\lambda, N) = (650, 2048)$ in blue, $(630, 4096)$ in red, and $(840, 4096^3)$ in black.

Most simulations in the literature that were aimed to a Reynolds number as high as possible for the given computational resources available [16, 17, 18] have used a grid spacing Δx that is about twice the Kolmogorov scale (η), which nominally is a measure of the size of the smallest scales in the flow. This resolution is adequate for low order statistics but not at high orders when fluctuations at sub-Kolmogorov scale size become important. In particular, recently theoretical work [19] has suggested that the resolution criterion in usual practice is not sufficiently accurate to capture all the details of intermittent quantities [8]. We illustrate this point in Fig. 7 where we show PDFs of velocity increments $\Delta_r u$ at different scale sizes r. The spatial separations are taken as either along (longitudinal, top of figure) or perpendicular to the velocity components (transverse, bottom of figure). At large scales (i.e. large values of r/η) both longitudinal and transverse increments are close to Gaussian. As r decreases (opposite to the directions of arrows drawn) wide tails develop which represent finite (albeit low) probability of very large fluctuations, i.e. show the property of intermittency. If a grid spacing of 2 Kolmogorov scales were sufficient to resolve the velocity gradients then in both parts (a) and (b) of this figure the two outermost data curves should effectively coincide. Yet a systematic difference is noticeable, which confirms the necessity of re-examining certain aspects of prior results in the literature — in particular, by using new simulations where advancing computer power is used to achieve increased accuracy at the small scale instead of increasing the Reynolds number.

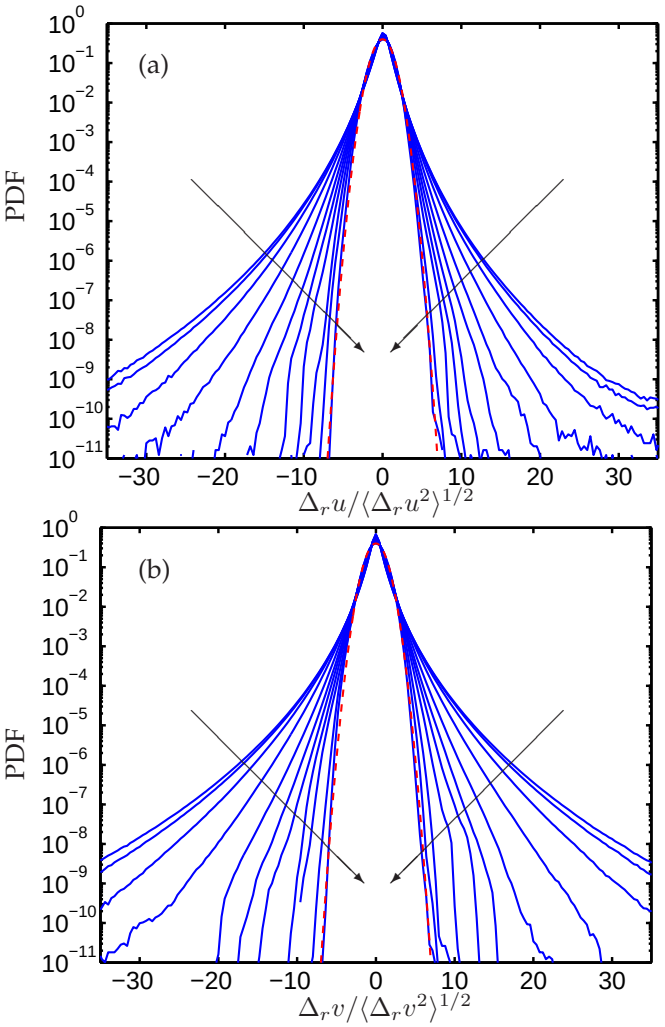

Figure 7: Probability density function of (a) longitudinal and (b) transverse velocity increments, from new Simulation II at $R_\lambda \approx 630$ (4096^3) normalized by their standard deviations $\langle\Delta_r u^2\rangle^{1/2}$ and $\langle\Delta_r v^2\rangle^{1/2}$. Arrows show increasing r/η from $r/\eta \approx 1$ to $r/\eta \approx 1024$ in powers of 2. Dashed red line: standardized Gaussian.

4 Summary and Discussion

We have described the formulation, implementation and performance of a new Fourier pseudo-spectral code based on two-dimensional domain decomposition for solving the Navier-Stokes equations on a periodic domain using the advanced computers with a high degree of parallelism. High scalability has been demonstrated on multiple machines even though some tuning is required (e.g., to determine the optimal processor grid geometry) depending on system architecture and network topology. On Ranger at TACC, weak and strong scalability above 90% has been observed at the largest cores counts currently available to user jobs. It is important, however, to mention that current timings can be improved yet further. In fact, our estimation of operation count (based on flop-counting software available on another machine) is $147N^3 \log_2 N$ per step which implies an aggregate 5.5 Tflops/s for $(N, M) = (4096, 16384)$ for which the best timing recorded to-date is about 22 secs. This corresponds to a performance slightly below 5% of peak theoretical value for the number of cores used. With help from TACC consultants, a greater understanding of both hardware and software details in the system environment, and detailed execution profiling tools to be made available, we believe further improvements in performance are likely in the future.

On the science side, we have provided preliminary results from a pair of 4096^3 simulations of isotropic turbulence, aimed separately at increasing the Reynolds number, and at resolving the small scale motions better than the standard usually achieved in prior work in the field. Results on energy spectrum and structure functions provide clear evidence of a wider inertial range, significantly more than one decade, that will be of great value in future studies of turbulent mixing and dispersion at higher Reynolds number than before. Intermittency and departures from classical Kolmogorov scaling are also clearly observed in higher-order structure functions and probability density functions of energy dissipation rate, enstrophy, as well as longitudinal and transverse velocity gradients. The new results give strong support to conclusions (both regarding Reynolds number effects and small-scale resolution) reached in a very recent paper [8] which ventured into parameter regimes not studied or discussed in the literature before. Since the simulation time is still short (about one large-eddy time scale) at the time of this writing, the statistical results reported here are subject to change as the simulations continue to progress and the sampling accuracy improves accordingly. However, the basic trends are likely to hold.

As for our prior simulation databases, we intend to archive the data and make them available to other researchers for analysis, using the Storage Resource Broker (SRB) software interface which is available at SDSC and other TeraGrid sites. We are also working towards a project goal of open-source code development which will be shared with interested members of the research community. Future simulations will, over time and together with our other project partners, be expanded in scope to include flows in more complex geometries (e.g. channel flow), and/or other physical processes such as chemical reactions or density stratification due to buoyancy effects.

Acknowledgments

We gratefully acknowledge support from NSF (Fluid Dynamics and PetaApps programs, Grants CBET-0554867 and OCI-0749223) and the TeraGrid in the form of advanced computing resources at multiple sites. Special thanks are due to Drs. W. Barth and K.W. Schulz of TACC for their expert assistance, and to Dr. A. Majumdar of SDSC for his friendly advice. Finally we thank Professor K.R. Sreenivasan, a leading member of the turbulence community, for his active involvement in the analysis and interpretation of our simulation data.

References

[1] K. R. Sreenivasan. Fluid turbulence. *Rev. Mod. Phys.*, 71:s383–s395, 1999.

[2] J. L. Lumley and A. M. Yaglom. A century of turbulence. *Flow Turb. Combust.*, 66:241–286, 2001.

[3] P. Moin and K. Mahesh. Direct numerical simulation: A tool in turbulence research. *Annu. Rev. Fluid Mech.*, 30:539–578, 1998.

[4] J. Jiménez. Computing high-reynolds-number turbulence: will simulations ever replace experiments? *J. Turb.*, 4:022, 2003.

[5] A. N. Kolmogorov. Local structure of turbulence in an incompressible fluid for very large Reynolds numbers. *Dokl. Akad. Nauk. SSSR*, 30:299–303, 1941.

[6] Z. Warhaft. Passive scalars in turbulent flows. *Annu. Rev. Fluid Mech.*, 32:203–240, 2001.

[7] P. K. Yeung. Lagrangian investigations of turbulence. *Annu. Rev. Fluid Mech.*, 34:115–142, 2002.

[8] D. A. Donzis, P. K. Yeung, and K. R. Sreenivasan. Dissipation and enstrophy in isotropic turbulence:

resolution effects and scaling direct numerical simulations. *Phys. Fluids*, 20:045108, 2008.

[9] K.R. Sreenivasan and R.A. Antonia. The phenomenology of small-scale turbulence. *Annu. Rev. Fluid Mech.*, 29:435–372, 2007.

[10] C. Canuto, M. Hussaini, A. Qarteroni, and T. Zang. *Sectral Methods in Fluid Dynamics*. Springer, 1988.

[11] R. S. Rogallo. Numerical experiments in homogeneous turbulence. NASA TM 81315, NASA Ames Research Center, Moffett Field, CA., 1981.

[12] P. K. Yeung, D. A. Donzis, and K. R. Sreenivasan. High-Reynolds-number simulation of turbulent mixing. *Phys. Fluids*, 17:081703, 2005.

[13] P. K. Yeung, S. B. Pope, A. G. Lamorgese, and D. A. Donzis. Acceleration and dissipation statistics of numerically simulated isotropic turbulence. *Phys. Fluids*, 18:065103, 2006.

[14] P. K. Yeung, S. B. Pope, E. A. Kurth, and A. G. Lamorgese. Lagrangian conditional statistics, acceleration and local relative motion in numerically simulated isotropic turbulence. *J. Fluid Mech.*, 582:399–422, 2007.

[15] http://www.sdsc.edu/us/resources/p3dfft.php.

[16] T. Gotoh, D. Fukayama, and T. Nakano. Velocity field statistics in homogeneous steady turbulence obtained using a high-resolution direct numerical simulation. *Phys. Fluids*, 14:1065–1081, 2002.

[17] Y. Kaneda, T. Ishihara, M. Yokokawa, K. Itakura, and A. Uno. Energy dissipation rate and energy spectrum in high resolution direct numerical simulations of turbulence in a periodic box. *Phys. Fluids*, 15:L21–L24, 2003.

[18] T. Watanbe and T. Gotoh. Inertial-range intermittency and accuracy of direct numerical simulations for turbulence and passive scalar turbulence. *J. Fluid Mech.*, 590:117, 2007.

[19] V. Yakhot and K. R. Sreenivasan. Anomalous scaling of structure functions and dynamic constraints on turbulence simulations. *J. Stat. Phys.*, 121:823, 2005.