

Features:

A Grand Challenge DNS at BSC

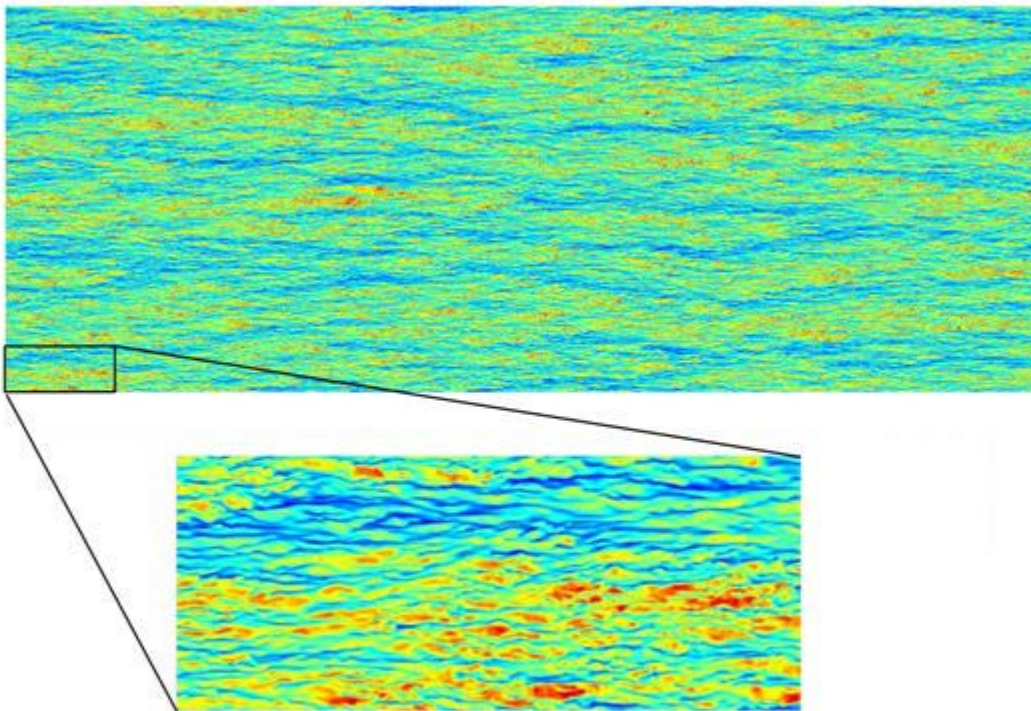
by Sergio Hoyas and Javier Jiménez Sendín
School of Aeronautics, Universidad Politécnica

The main goal of this grand challenge application is to better understand the physics of turbulent boundary layers at high Reynolds numbers. This simulation was performed on the Barcelona Supercomputing Center's supercomputer MareNostrum and is now the world's largest simulation in the field of wall turbulence, with $2e10$ grid points.

It is important to note the multi-scale character of the intermediate layer present in turbulent flows. This is where the longest flow structures are found and where the range of scales is widest. The purpose of this experiment is to extend available simulations to a Reynolds number twice as high as had been previously studied. Wall-bounded turbulence has been studied for a long time, but it is only recently -- in fact, only with this experiment -- that we are able to represent the logarithmic layer linking the near-wall and outer flow regions. This is the location of the self-similar energy and momentum cascades, which are the essence of turbulent flows.

Based on previous simulations and experiments done by other researchers, we have some understanding of the nonlinear dynamics of the near-wall structures and the largest outer scales. Both are essentially single-scale phenomena. However, the dynamics of the intermediate range of scales are more poorly understood because it has been impossible up to now to simulate a Reynolds number large enough for a multi-scale range to be available.

The problem is not just of scientific interest. Wall-bounded turbulence is the interface between the ambient fluid, water or air, and moving vehicles, and between flows and pipes and channels. It is also a key component of the atmospheric boundary layer. The large scales, for example, play an important role in the dispersion of chemical agents in the atmosphere. Due to their associated low frequencies, their pressure fluctuations are responsible for undesirable aerodynamic loads which may result in structural fatigue and long-range noise in vehicles. More than half of the friction drag in vehicles resides in the near-wall and intermediate layers. The hopes of controlling and alleviating these phenomena reside with the understanding of their dynamics.



This picture symbolizes the smallest scale representing the flow.

"Streaks in the streamwise component of velocity at the viscous sublayer."

The program we have developed performs a turbulence simulation using the Navier-Stokes equations for an incompressible fluid in a plane channel between two infinite parallel plates. No modeling is used. The computation is carried out in a doubly periodic domain in two wall-parallel directions, which must be large enough to minimize artifacts due to spurious periodicity. From our experience with lower Reynolds numbers we have chosen a domain of size $25h$ in the streamwise direction (x) and $10h$ in the spanwise direction (z) where h is the channel half-thicknesses. This procedure, although both memory- and compute-intensive, allows us to compute any property of the flow with greater accuracy than would be obtainable using experimental measurements in laboratory flows. In particular, it allows us to access variables, such as time- and space-resolved velocity gradients and vorticity, which cannot be measured in the laboratory. The Reynolds number used in the simulation is comparable to those used in many laboratory flows and higher than those used in most of the wall-bounded turbulent flows that have been measured in any detail beyond mean and fluctuating velocities.

The program code is based on previous versions developed by our group during the past fifteen years and broadly follows the standard spectral code developed at the end of the 1980s by groups at Göttingen and at NASA Ames. The version used in our previous simulations has been widely distributed and is now used at the University of TX at Austin and Nagoya University in Japan, for example. For the present simulation, a newer version that removes several limitations of the older code has been developed, saving up to fifty percent of the memory requirements and improving code flexibility. For example, the older version could only be run efficiently in a number of nodes that was a divisor both of the number of computational planes in the wall-normal (y) direction, and of those in the streamwise (x) direction. The former is usually an order of magnitude smaller than the latter, and limited the use of that code to relatively small

machines. That limitation has been removed from the present version. The new code also uses a higher-order Runge-Kutta time-stepper, allowing a longer time step, and a 'spectral' sixth-order Compact Finite Differences (CFD) discretization in the y-direction, which allows greater flexibility in the choice of the grid. Those two changes save about a factor of two in computer time.

The code uses Fourier discretization in x-z directions and CFD in the y direction, with $m_x * m_y * m_z$ points (6144 * 633 * 4608 in this case). For each sub-step of R-K scheme, the data set is distributed in y-z planes. To avoid load imbalances, the number of processors must be divisible by the number of complex planes, ($m_{galx}/3$ after dealising, 2048 in the present case). While in this memory organization, the data is transformed from Fourier space in z to physical space in z. This is needed to compute the nonlinear terms of the Navier-Stokes equations.

After this, the data set is moved from y-z planes to x-lines using MPI routines. Then the data is transformed to physical space in x, and the nonlinear terms are computed using the 3/2 technique to avoid aliasing. The nonlinear term is then transformed back to x-Fourier, moved to y-z planes and transformed to z-Fourier. We then solve the viscous problem and move to the next sub step of the R-K. The two global (all to all) transposes needed in this step constitute most of the communication cost of the code.

In order to run this code in a supercomputer with more than 2048 processes with optimal performance, we did some specific optimization, such as tuning the code to take advantage of the vectorial instruction set of the PPC970-FX CPU. This was achieved using a version of the FFTW library provided by the Barcelona Supercomputing Center's Deep Computing research group. In addition, the most expensive I/O operations were assigned to dedicated processors to reduce the global file system contention. The latter approach allowed us to overlap the computing stage of 2048 processors with the massive writing operations, using only 60 extra processors. The aim was that the code was able to follow the normal execution while the dedicated processors saved to disk the huge amount of information produced by the code, up to 25 TB for the entire simulation.

The MareNostrum facility became the optimal platform to run this grand Challenge simulation because of the project's computational needs, together with the large amount of memory required per process and the huge amount of data interchanged, roughly 122 EB. It was a challenge for the support team of MareNostrum because of the simulation's size. MareNostrum is a facility available to the global scientific community.

In summary, using close to five million CPU-hours (more than 570 years in a single computer), we performed a landmark simulation in wall turbulence. While the full analysis of the results from this simulation will take several years, some results are already open to the community via the website <http://torroja.dmt.upm.es>.

For more information, visit the Barcelona Supercomputing Center website at www.bsc.es.

You can contact Sergio Hoyas at sergio@torroja.dmt.upm.es and Javier Jiménez Sendín at jimenez@torroja.dmt.upm.es.