

# Direct numerical simulation of a neutrally stratified turbulent Ekman layer

Roderick Johnstone, Mike Ashworth, CCLRC Daresbury Laboratory, Gary Coleman, Aeronautics and Astronautics, University of Southampton

The atmospheric planetary boundary layer (PBL) is the region of the atmosphere directly affected by fluxes of momentum, heat, and mass to and from the surface, responding to surface forcings within a short timescale of an hour or less.

Above the boundary layer, the pressure gradient and Coriolis force are in approximate balance, while the velocity is normal to both (this is called geostrophic flow).

As the surface is approached, however, the wind speed decreases to zero, and with it the effects of the Coriolis force; this causes the wind to veer in the direction of low pressure.

In the absence of turbulence, it is possible to obtain an analytical solution to the governing Navier-Stokes equations for this flow; this is the Ekman spiral, shown in Figure 1 as the 'Laminar' solution. However, the PBL is, in reality, turbulent. The 'Turbulent' solution in Figure 1 is a spatially and temporally averaged result from a simulation of the turbulent Ekman layer currently running on HPCx (described below).

The PBL is important for weather prediction – firstly, almost all interaction between people and the atmosphere takes place there. There is therefore a demand for forecasts of near-surface conditions such as wind speed and temperature. Conversely, surface forcings of the atmosphere involve the PBL; for instance, it is from the surface that the atmosphere gains moisture, and most of its solar heat input; the surface also exerts drag on the atmosphere. Turbulence affects all of these fluxes.

Unfortunately, it is not feasible to resolve PBL turbulence in weather prediction models (which instead model the PBL), nor even to simulate it numerically at its real Reynolds number ( $Re$ ). This is because the computational expense of such simulations scales approximately with the cube of the Reynolds number, and  $Re$  is very large indeed in geophysical flows (up to about  $10^8$  for the PBL).

However, it is possible to use direct numerical simulation (DNS) of the turbulent Ekman layer at a lower  $Re$  than that of the PBL, in order to test parametrizations of the PBL, and to obtain values for the empirical constants they require. Idealized simulations are in fact quite suitable as it is possible to exclude from them effects that the theory is not designed to represent (which is difficult to do when using field data for the same purpose).

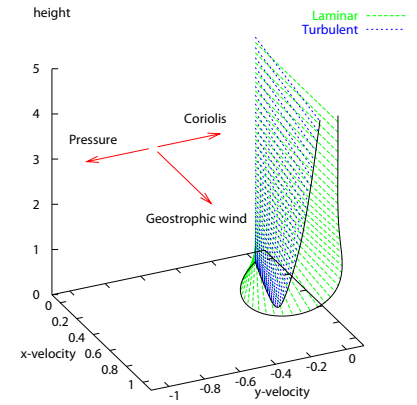


Figure 1: Laminar and turbulent Ekman boundary layer.

A simulation of this kind is currently being run on HPCx. The results of this work could be used to improve subgrid-scale models used in Large-Eddy Simulations, and to test similarity theories of the turbulent Ekman layer; the principal aim is in fact to use the results obtained in combination with the theory of Spalart [1].

This theory extends the similarity theory of Csanady [2] to low Reynolds numbers, which makes it possible to extrapolate results from a DNS to the much higher  $Re$  of the PBL. The collection of data at a series of increasing Reynolds numbers will permit greater confidence to be placed in such extrapolation.

Coleman et al [5,6], employed a code implementing the method of Spalart et al [3,4], for the simulation of the neutrally-stratified, incompressible, turbulent Ekman boundary layer at Reynolds numbers of 400, 500 and 1000, based on the geostrophic wind speed and viscous Ekman layer depth. This work is now being extended with a further simulation at  $Re=2000$ , using HPCx.

This code was originally written in the Vectoral language [7] and run on the Cray C90. Since then it has been translated to Fortran 95 and parallelised using MPI. Because different transforms are applied in the vertical and horizontal directions, it makes sense to decompose the domain alternately into vertical and horizontal planes for the purpose of parallelization. The global character of spectral discretisations makes heavy demands with regard to communication, however. Here, this takes the form of a transpose of large distributed arrays, requiring all-to-all communication.

The code has been optimised to improve the performance. The following were found to be beneficial:

1. Use of dgemm from BLAS for matrix multiplication. Obvious as this is, it is worth a 70% reduction in run time in current production use.
2. Use of the ESSL routine zgetmo to transpose arrays for the FFTs (about 7% reduction in run time).
3. MPI\_Alltoallv is used to simultaneously perform the parallel transpose together with a broadcast of some additional data. Separating the broadcast from the all-to-all creates an opportunity to overlap the all-to-all communication with some computation; this was attempted, but found to be slower than simply amalgamating the two.

*Continued opposite.*

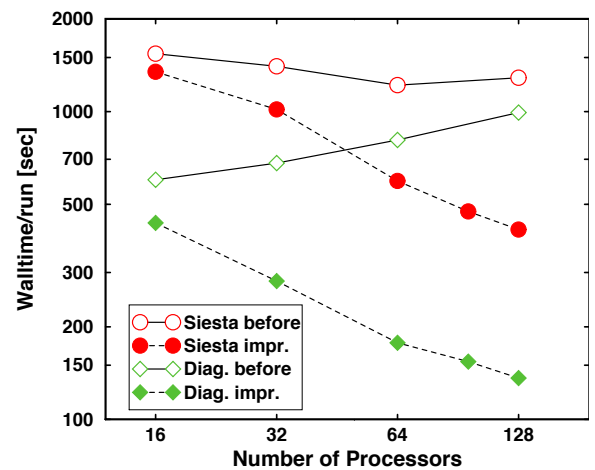
# Improved parallel performance of SIESTA for the HPCx Phase2 system

Joachim Hein,  
HPCx Terascaling Team

SIESTA is a materials code using self-consistent density functional theory. The code aims at the simulation of materials with a large number of atoms. The name SIESTA is an acronym for 'Spanish Initiative for Electronic Simulations with Thousands of Atoms'. SIESTA uses order-N algorithms allowing linear scaling with the number of atoms [1].

The application is used by several users of the HPCx service. In particular, the e-Minerals consortium is investigating the environment from the molecular level. The science being performed on HPCx is described in the article on p10.

To obtain acceptable runtimes on materials with such a large number of atoms, good parallel scalability is required. Benchmarking by the HPCx terascaling team showed that the performance of the diagonaliser was the key obstacle for the code to scale to a large number of processors. Furthermore, using a larger problem size did not help the parallel performance of the diagonaliser. The performance was clearly inferior when compared to the experience the HPCx terascaling team has gained with eigensolvers [2]. Redistributing the data inside the diagonaliser leads to a dramatically improved performance when using a large number of processors. For our benchmarking configuration SIESTA now runs up to three times faster when used on 128 processors. With the recent hardware upgrade and the change



Performance of Siesta and its Diagonaliser for a benchmark system before and after the improvement

from 8-way to 32-way nodes, this is a very timely improvement. SIESTA is now able to use several 32-way SMP-nodes for a single calculation. These improvements will be incorporated in future SIESTA releases.

SIESTA benefits further from the upgrades to the switch micro-code, introduced to the service in July 2004. On 128 processors we observed an improvement of about 14%.

A full technical report on this work, containing detailed instructions on recommended environment settings is available [3].

## References:

- [1] J.M. Soler, et al., J. Phys.: Condens. Matter 14, 2745 (2002).
- [2] Elena Breitmoser, Andy Sunderland, 'A performance study of PLAPACK and ScaLAPACK Eigensolvers on HPCx for the standard problem', HPCx technical report HPCxTR0406
- [3] Joachim Hein, 'Improved parallel performance of SIESTA for the HPCx Phase2 system', HPCx technical report HPCxTR0410

4. The recent HPS switch microcode upgrade (SP7) which took place on 28th July 2004 has markedly improved the parallel scaling of this code on HPCx (see figure 2).

This simulation is funded by EPSRC and managed through the UK Turbulence Consortium.

## References:

- [1] Spalart, P.R., 1989, 'Theoretical and numerical study of a three-dimensional boundary layer', Journal of Fluid Mechanics 205, pp 319-340
- [2] Csanady G.T., 1967, 'On the 'resistance law' of a turbulent Ekman layer', Journal of the Atmospheric Sciences 24, pp 467-471
- [3] Spalart, P.R., 1986, 'Numerical Simulation of Boundary Layers: Part 1. Weak Formulation and Numerical Method', NASA Technical Memorandum 88222
- [4] Spalart, P.R., Moser, R.D., and Rogers, M.M., 1991, 'Spectral Methods for the Navier-Stokes Equations with One Infinite and Two Periodic Directions', Journal of Computational Physics 96, pp297-324

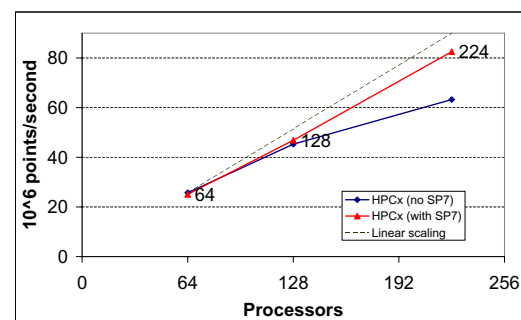


Figure 2: Performance of the Turbulent Ekman Boundary Layer code before and after the HPS switch microcode upgrade (SP7).

- [5] Coleman, G.N., Ferziger J.H., and Spalart, P.R., 1990, 'A numerical study of the turbulent Ekman layer', Journal of Fluid Mechanics 213, pp 313-348
- [6] Coleman, G.N., 1999 'Similarity Statistics from a Direct Numerical Simulation of the Neutrally Stratified Planetary Boundary Layer', Journal of the Atmospheric Sciences 56, pp 891-900
- [7] Wray, A. 2002, <http://merrimac.stanford.edu/brook/vectoral.pdf>