11B.5 A HIGHLY PARALLEL ALGORITHM FOR TURBULENCE SIMULATIONS IN PLANETARY BOUNDARY LAYERS: RESULTS WITH MESHES UP TO 1024^3

Peter P. Sullivan[*] and Edward G. Patton

National Center for Atmospheric Research, Boulder, CO

1. INTRODUCTION

Petascale computing (*e.g.,* UCAR/JOSS, 2005) has the potential to alter the landscape of turbulence simulations in planetary boundary layers (PBLs). Increased computer power using $O(10^4 - 10^5)$ or more processors will permit large-eddy simulations (LESs) of turbulent flows over a wide range of scales in realistic outdoor environments, for example, flow over hills, atmosphere-land interactions (Patton et al., 2005), boundary layers with surface water wave effects (Sullivan et al., 2008, 2007), and weakly stable nocturnal flows (Beare et al., 2006) to mention just a few. However, computational algorithms need to evolve in order to utilize the large number of processors available in the next generation of machines. Here we briefly describe some of our recent developments focused on constructing a massively parallel large-eddy simulation (LES) code for simulating incompressible Boussinesq atmospheric and oceanic boundary layers. The performance of the code is evaluated on varying meshes utilizing as many as 16,384 processors. As an application, the code is used to examine the convergence of LES solutions for a daytime convective PBL on grids varying from 32^3 to 1024^3.

2. 2-D DOMAIN DECOMPOSITION

Typical LES model equations for dry Boussinesq boundary layers include at a minimum: a) transport equations for momentum $\rho\mathbf{u}$; b) a transport equation for a conserved buoyancy variable (*e.g.,* virtual potential temperature θ_v); c) a discrete Poisson equation for a pressure variable π to enforce incompressibility; and closure expressions for subgrid-scale (SGS) variables, *e.g.,* a subgrid-scale equation for turbulent kinetic energy e. In our LES code these equations are integrated forward in time using a fractional step method. The spatial discretization is second-order finite difference in the vertical direction and pseudospectral in horizontal planes (Moeng, 1984). Dynamic time stepping utilizing third-order Runge-Kutta with a fixed Courant-Fredrichs-Lewy

[*]*corresponding author address:* Peter P. Sullivan, National Center for Atmospheric Research, P. O. Box 3000, Boulder, CO 80307-3000; *email*: pps@ucar.edu

(CFL) number (Sullivan et al., 1996; Spalart et al., 1991) is employed. Evaluating horizontal derivatives with Fast Fourier transforms (FFTs) and solving the elliptic pressure equation are non-local operations which impact the parallelization.

Our previous code parallelizes the flow model described above using a single domain decomposition procedure that combines distributed memory MPI tasks (Aoyama and Nakano, 1999) and shared memory OMP threads (Chandra et al., 2001). The full computational domain is naturally first decomposed in the vertical z direction using MPI, *i.e.,* a subset of vertical levels and full horizontal $x - y$ domains are assigned to each computational node. Work on a node is then further partitioned amongst local threads using OMP directives. This scheme has some advantages; 1) it does not split FFTs across spatial directions since threads share the same memory and thus a specialized parallel FFT package is not required; and 2) it can utilize the architecture of machines with large numbers of processors per computational node (*e.g.,* the IBM SP5 with 16 processors/node). However the scheme is limited on computing platforms which have few processors/node (*e.g.,* the Cray XT4 with 2 processors/node), and moreover we find the OMP directives require continual maintenance that adds overhead and complexity.

To streamline the code and increase its flexibility a new parallel algorithm is designed based on the following criteria: 1) accomplish 2-D domain decomposition using solely MPI parallelization; 2) preserve pseudospectral (FFT) differencing in $x - y$ planes; and 3) maintain a Boussinesq incompressible flow model. The ability to use 2-D domain decomposition has been shown to be a significant advantage in pseudospectral simulation codes as it allows direct numerical simulations of isotropic turbulence on meshes of 2048^3 or more (Pekurovsky et al., 2006). A sketch of the domain decomposition layouts that adhere to our constraints is given in figure 1. We mention 2-D domain decomposition in $x - y$ planes is compatible with the use of low-order finite difference schemes (Raasch and Schröter, 2001) and mesoscale codes that adopt compressible equations (Michalakes et al., 2005).

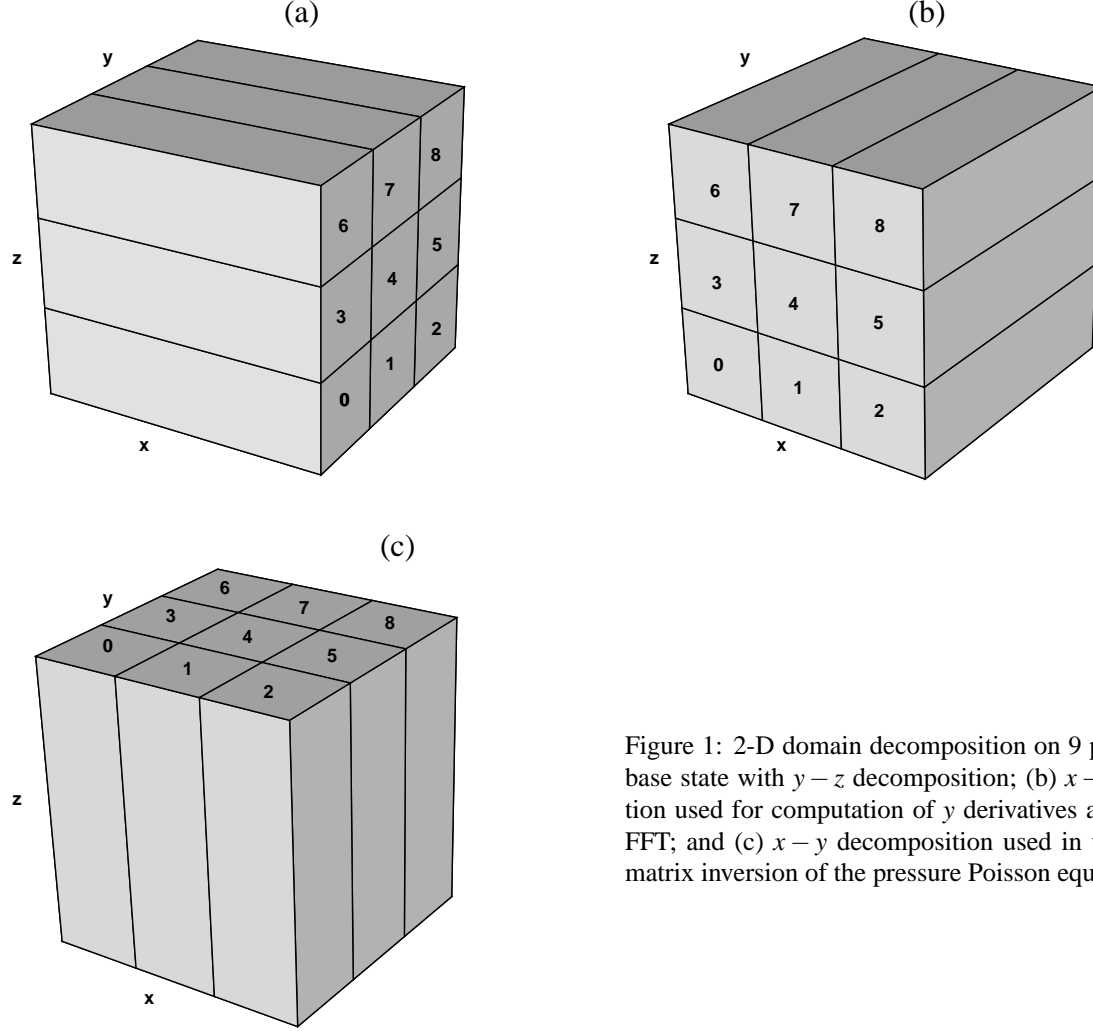In our 2-D domain decomposition, each processor op-

(a)

(b)

(c)

Figure 1: 2-D domain decomposition on 9 processors: (a) base state with $y - z$ decomposition; (b) $x - z$ decomposition used for computation of y derivatives and 2-D planar FFT; and (c) $x - y$ decomposition used in the tridiagonal matrix inversion of the pressure Poisson equation.

erates on constricted three-dimensional "bricks or pencils" sub-sampled in x, y or z directions. Brick-to-brick communication is a combination of transposes and ghost point exchange. To preserve pseudospectral differencing in the horizontal directions a custom MPI matrix transpose was designed and implemented. Note other non-local schemes, *e.g.,* compact finite difference (Lele, 1992) or fully spectral direct numerical simulation codes (Werne and Fritts, 1999), require similar communication patterns. Our transpose routines perform the forward and inverse operations

$$
f(x,y,z) \begin{bmatrix} \text{all } x \\ y_s \leq \quad y \leq y_e \\ z_s \leq \quad z \leq z_e \end{bmatrix} \iff
$$

$$
f^T(y,x,z) \begin{bmatrix} \text{all } y \\ x_s \leq \quad x \leq x_e \\ z_s \leq \quad z \leq z_e \end{bmatrix} \quad (1)
$$

on a field f using a subset of horizontal processors as

shown in figure 1a and 1b. In (1) and following equations, subscripts $(\ )_{s,e}$ denote starting and ending locations in the (x,y,z) directions. The data transpose shown schematically in figure 1a and 1b only requires *local* communication, *i.e.,* communication between processors in groups $[0,1,2]$, $[3,4,5]$, and $[6,7,8]$. Derivatives $\partial f / \partial y$, which are needed in physical space, are computed in a straightforward fashion using the sequence of steps:

1. forward x to y transpose $f \to f^T$;

2. FFT derivative $\partial f^T / \partial y$; and ,

3. inverse y to x transpose $\partial f^T / \partial y \to \partial f / \partial y$.

Existing serial 1-D FFT routines for real and complex arrays are used as in previous implementations. Note with this algorithm so-called ghost points used in computing derivatives $\partial f / \partial z$ are only needed on the top and bottom faces of each brick in figure 1a.

The 2-D brick decomposition of the computational domain also impacts the pressure Poisson equation solver.

In an incompressible Boussinesq fluid model the pressure π is a solution of the elliptic equation

$$\nabla^2 \pi = r, \qquad (2)$$

where the source term r is the numerical (discrete) divergence of the unsteady momentum equations (*e.g.*, Sullivan et al., 1996). The solution for π begins with a standard forward 2-D Fourier transform of (2):

$$- \left(k_x^2 + k_y^2 \right) \hat{\pi} + \frac{\partial^2 \hat{\pi}}{\partial z^2} =$$

$$\hat{r}(k_y, k_x, z) \ \text{with} \ \begin{bmatrix} \text{all } k_y \\ k_{xs} \leq k_x \leq k_{xe} \\ z_s \leq z \leq z_e \end{bmatrix}, \quad (3)$$

where (k_x, k_y) are horizontal wavenumbers. At this stage the data layout on each processor is as shown in figure 1b. Next, custom routines carry out forward k_y to z and inverse z to k_y matrix transposes on the source term of (3):

$$\hat{r}(k_y, k_x, z) \begin{bmatrix} \text{all } k_y \\ k_{xs} \leq k_x \leq k_{xe} \\ z_s \leq z \leq z_e \end{bmatrix} \Longleftrightarrow$$

$$\hat{r}^T(z, k_x, k_y) \begin{bmatrix} \text{all } z \\ k_{xs} \leq k_x \leq k_{xe} \\ k_{ys} \leq k_y \leq k_{ye} \end{bmatrix} \quad (4)$$

Again notice the communication pattern needed to transpose from figure 1b to 1c is accomplished locally by processors in groups $[0, 3, 6]$, $[1, 4, 7]$, and $[2, 5, 8]$. The continuous storage of $\hat{r}^T$ along the z direction allows straightforward tridiagonal matrix inversion for pairs of horizontal wavenumbers on each processor. This step is repeated for all pairs of horizontal wavenumbers and provides the transposed field $\hat{\pi}^T(z, k_{xs} : k_{xe}, k_{ys} : k_{ye})$. To recover the pressure field in physical space we retrace our steps: $\hat{\pi}^T \rightarrow \hat{\pi}$ followed by an inverse 2-D Fourier transform $\hat{\pi} \rightarrow \pi$. In designing the present algorithm, we also considered using the parallel tridiagonal solver described by Gibbs (2004) for the solution of the Poisson equation but found it not well suited for the present scheme.

With these enhancements our new algorithm allows very large number of processors $O(10^4)$ to be utilized. An important feature of the algorithm is that no global MPI ALLTOALL communication between processors is required. Instead, the MPI routine SENDRECV is wrapped with FORTRAN statements to accomplish the desired communication pattern. The scheme outlined above introduces more communication but the messages are smaller and hence large numbers of gridpoints can be used. Also, the total number of processors is not limited by the number of vertical gridpoints. For example, this
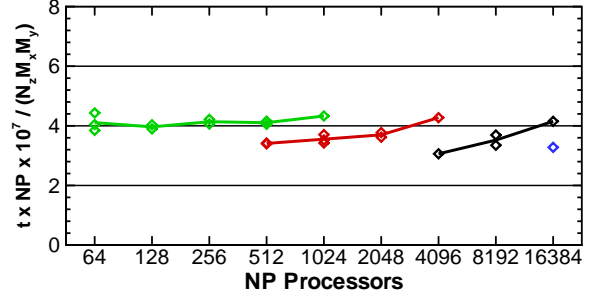

Figure 2: Computational time per gridpoint for different combinations of problem size and 2D domain decomposition for the Cray XT4 (an example of strong scaling). a) green lines and symbols problem size 512^3; b) red lines and symbols 1024^3; c) black lines and symbols 2048^3; and d) blue symbol 3072^3. For a given number of total processors NP the symbols are varying vertical and horizontal decompositions, *i.e.*, different combinations (NP_z, NP_{xy}).

flexibility allows simulations in boxes with large horizontal and small vertical extents. The transpose routines are general and allow arbitrary numbers of mesh points, although the best performance is of course realized when the load is balanced across processors. Single files, similar to FORTRAN direct access files, are written and read using MPI I/O (Gropp et al., 1998). We find MPI I/O makes the code robust across different machine architectures and simplifies the logic required for restarts, especially if the number of processors changes during the course of a simulation. Finally, the code is compliant with the FORTRAN-90 programming standard.

The performance of the code for varying workload as a function of the total number of processors NP is provided in figures 2 and 3 for 3 different machine architectures. $NP = NP_z \times NP_{xy}$ where NP_z and NP_{xy} are the number of processors in the vertical and horizontal directions, respectively. In each figure, the vertical axis is total computational time $t \times NP$ divided by total work. N_z is the number of vertical levels and $M_{x,y}$ is proportional to the FFT work, *i.e.*, $M_{x,y} = N_{x,y} \log N_{x,y}$ with $N_{x,y}$ the number of gridpoints in the x and y directions. Ideal scaling corresponds to a flat line with increasing number of processors. The timing tests illustrate the present scheme exhibits both strong scaling (problem size is held fixed and the number of processors is increased) and weak scaling (the problem size grows as the number of processors increases so the amount of work per processor is held constant) over a wide range of problem sizes and is able to use as many as 16,384 processors, *i.e.*, the maximum number available to our application on the Cray XT4. Further, the results are robust for varying combinations of (NP_z, NP_{xy}). Generally, the performance only begins
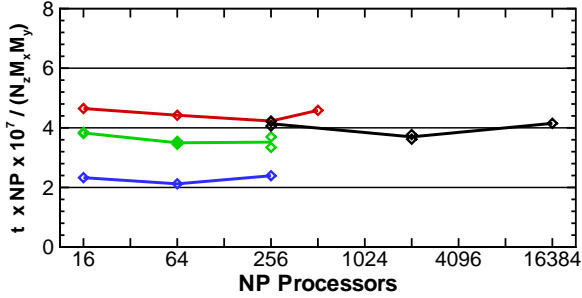
Figure 3: Computational time per gridpoint for a fixed amount of work per processor (an example of weak scaling). Red, green, and blue lines 60,000 points/processor for different machines. Cray XT4 red line; Dual core IBM SP5+ green line; Single core IBM SP5 blue line. Black lines and symbols 524,288 points/processor for Cray XT4. For a fixed number of total processors NP multiple symbols are different combinations of (NP_z, NP_{xy}).

Table 1: Simulation properties

Run	Gridpoints	z_i (m)	$z_i/\triangle z$	w_* (ms^{-1})
A	32^3	1120	17.5	2.06
B	64^3	1116	34.9	2.06
C	128^3	1123	70.2	2.06
D	256^3	1095	137.0	2.05
E	512^3	1088	272.0	2.04
F	1024^3	1066	536.7	2.04

to degrade when the number of processors exceeds about 8 times the smallest dimension in the problem owing to increases in communication overhead.

3. GRID SENSITIVITY

Parallel codes allow one to simulate PBLs with a wider range of scale interactions and external forcings, *e.g.,* Jonker et al. (1999) and Sullivan et al. (2007). Here, we briefly explore one aspect of this much larger issue, *viz.,* the sensitivity and convergence of LES solutions as the grid mesh is varied. Checking numerical convergence of LES solutions is not readily addressed in usual LES practice since the computational demands needed to carry out the required grid studies become prohibitive for a 3-D time dependent turbulent flow (*e.g.,* see LES intercomparison studies by Beare et al. (2006), Bretherton et al. (1999) and, Nieuwstadt et al. (1993)). A series of LES on a fixed computational domain with grid resolutions varying from 32^3 to 1024^3 are performed to examine the solution convergence and flow structures us-

ing the parallel algorithm described in Section 2. For each grid resolution, the mesh spacing is constant in the three (x, y, z) coordinate directions. A canonical daytime convective PBL is simulated in a computational domain $(X_L, Y_L, Z_L) = (5120, 5120, 2048)$ m. The PBL is driven by a constant surface heat flux $Q_* = 0.24$ K m s^{-1} and weak geostrophic winds $(U_g, V_g) = (1, 0)$ m s^{-1}. Other external inputs are surface roughness $z_o = 0.1$ m, Coriolis parameter $f = 1 \times 10^4$ s^{-1}, and initial inversion height $z_i \sim 1000$ m. The PBL is dominated by convection since the Monin-Obukhov length scale $L \leq -1.5$ m and thus the metric $z_i/L = O(-500)$ (Moeng and Sullivan, 1994). All simulations are started from small random seed perturbations in potential temperature near the surface. The simulations are carried forward for more than 25 large eddy turnover times $T = z_i/w_*$. The convective velocity scale $w_* = (gQ_*z_i/\theta_o)^{1/3}$ with g gravity and θ_o a reference potential temperature. See Moeng (1984), Moeng and Wyngaard (1989) and Sullivan et al. (1998) for a further description of the simulation design. Bulk properties of the simulations are given in Table 1.

4. PRELIMINARY RESULTS

4.1 Flow visualization

Fine mesh simulations allow a wider range of large and small scale structures to co-exist and thus interact in a turbulent flow. Flow visualization in figures 4 and 5 illustrates the formation of both large and small structures. In figure 4, we observe the classic formation of plumes in a convective PBL. Vigorous thermal plumes near the top of the PBL can trace their roots through the middle of the PBL down to the surface layer. Convergence at the common corners of the hexagonal patterns in the surface layer leads to the formation of strong updrafts which evolve into large scale plumes that fill and dominate the dynamics of the daytime PBL. Near the inversion a descending shell of motion readily develops around each plume.

Closer inspection of the large scale flow patterns in figure 4 also reveals coherent smaller scale structures. This is demonstrated in figure 5 where we track the evolution of 10^5 particles over about 1000 seconds. Over the limited region where the particles are released the flow is dominated by a persistent line of larger scale upward convection. On either side of the convection line descending motion develops. Near the surface these downdrafts turn laterally and converge. The outcome of this surface layer convergence spawns many small scale vertically oriented vortices, *i.e.,* dust devils. The rapidly rotating vortices are readily observed, persist in time, and rotate in both clockwise and counterclockwise directions. Often the vortices coalesce in a region where a
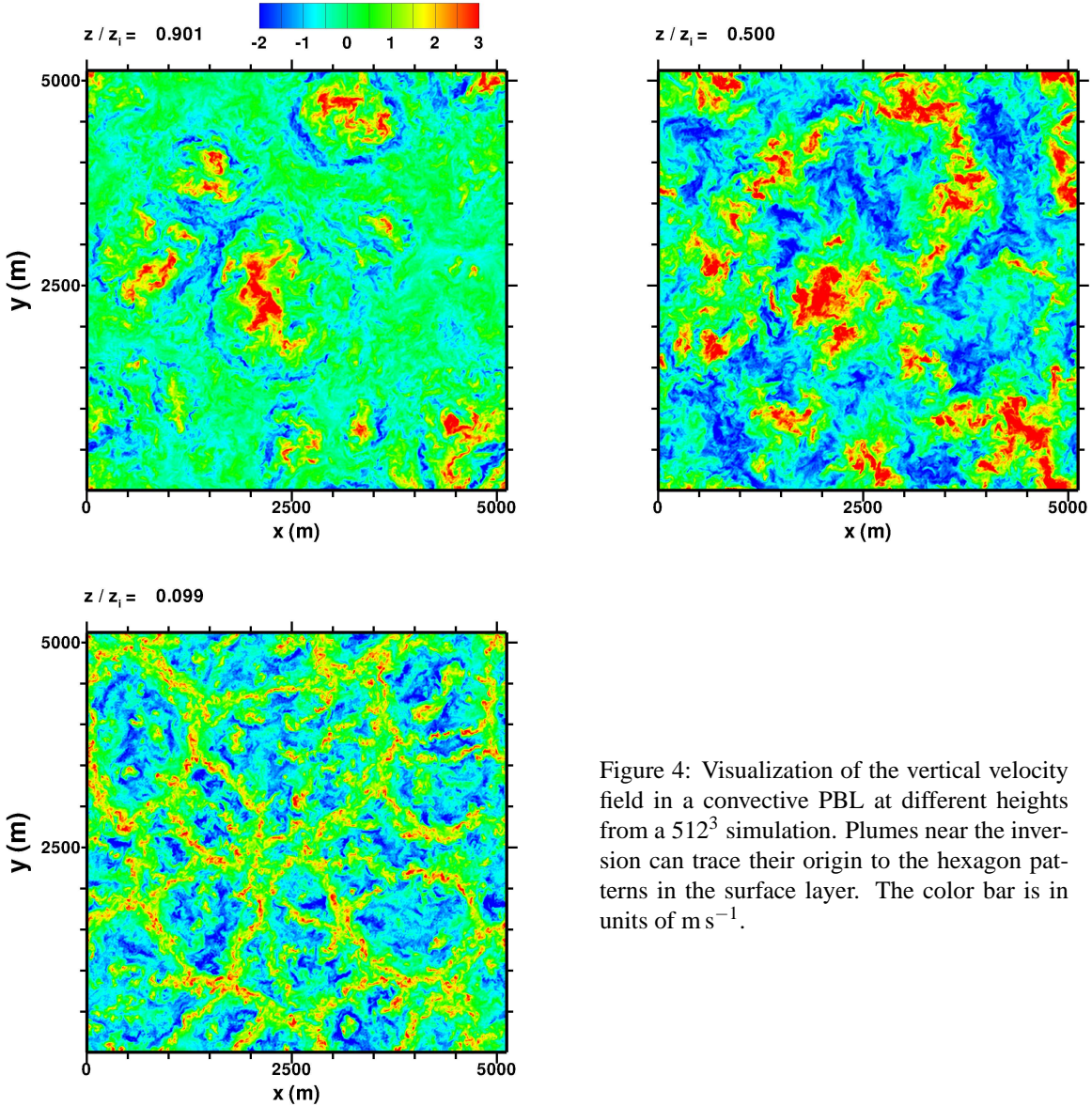
Figure 4: Visualization of the vertical velocity field in a convective PBL at different heights from a 512^3 simulation. Plumes near the inversion can trace their origin to the hexagon patterns in the surface layer. The color bar is in units of $m\,s^{-1}$.

coherent thermal plume erupts. Coarse mesh LES hints at these coherent vortices but fine resolution simulations allow a detailed examination of their dynamics within a larger scale flow. Previously, Kanak (2005) has observed the formation of dust devils in convective simulations, but in small computational domains $O(750)$ m.

4.2 Statistics

The impact of mesh resolution on typical (normalized) turbulence statistics, *viz.,* SGS dissipation ε, total turbulent kinetic energy E, and maximum vertical velocity w_{max}:

$$\varepsilon = \left[\frac{C_\varepsilon e^{3/2}}{\triangle_f}\right]\left[\frac{z_i}{w_*^3}\right], \qquad (5a)$$

$$E = \left[\frac{\overline{u}_i\overline{u}_i}{2} + e\right]\left[\frac{1}{w_*^2}\right], \qquad (5b)$$

$$w_{max} = \frac{|\overline{w}|_{max}}{w_*} \qquad (5c)$$

is shown in figure 6. In (5), the resolved scale velocity components are $\overline{u}_i = (\overline{u}, \overline{v}, \overline{w})$, the subgrid-scale energy $e = (\overline{u_iu_i} - \overline{u}_i\overline{u}_i)/2$, the LES filter width is $\triangle_f$, and $C_\varepsilon \sim 0.93$ is a modeling constant (Moeng and Wyngaard, 1988). A premise of LES, and also the basis of most SGS modeling, states that the average dissipation is constant
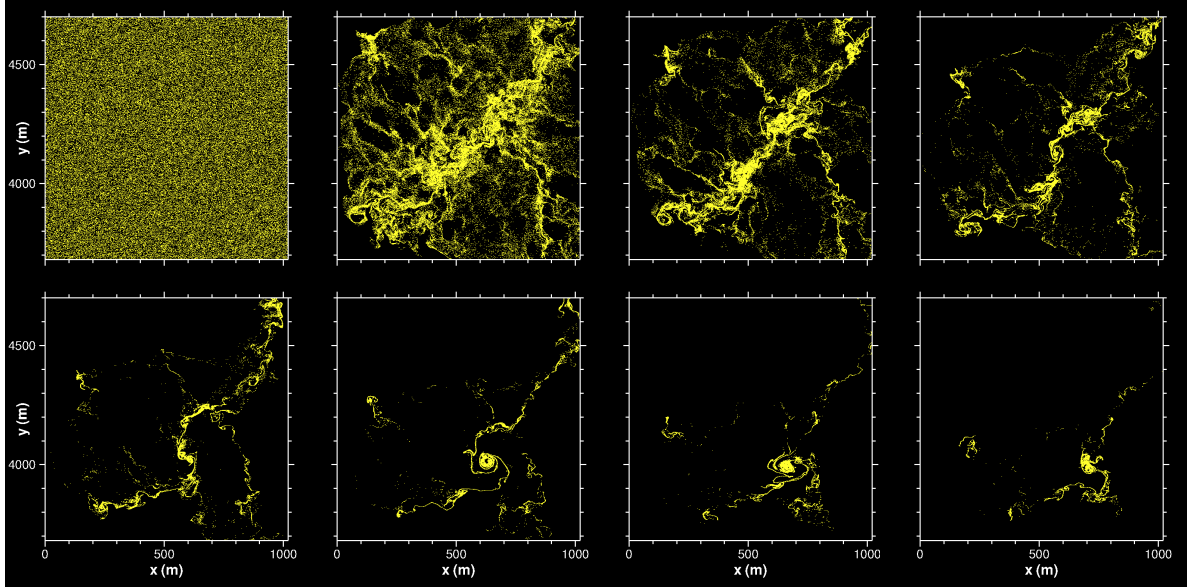
Figure 5: Visualization of particles released in a convective PBL at $z/z_i \sim 0.2$ over a limited horizontal extent from a 1024^3 simulation of convection. The viewed area is $\sim 3.8\%$ of the total horizontal domain. Notice the evolution of the larger scale line of convection into small scale vortical dust devils. Time advances from left to right beginning along the top row of images.

if the filter width lies in the inertial subrange range; then $e \sim \triangle_f^{2/3}$ (Moeng and Wyngaard, 1988). Similarly the total turbulent kinetic energy, *i.e.,* the sum of resolved and SGS pieces, also tends to a constant. Figure 6 is a test of this hypothesis. Notice the LES solutions converge over the bulk of the PBL when the mesh is 256^3 or greater. In other words, these low-order LES statistics, for this particular convective PBL, become independent of the grid resolution only when $z_i/\triangle_f > 130$. This is typically finer resolution than is used in routine calculations of free convection. It is encouraging to see only small changes when the resolution is increased from 512^3 to 1024^3. The above results hint that the SGS model impacts the coarse solutions in important ways, especially when the filter width approaches the energy containing scales (Sullivan et al., 2003).

Moeng and Rotunno (1990) identify the vertical velocity skewness S_w as a critical parameter in boundary layer dynamics. In convective PBLs, S_w is an indicator of the updraft-downdraft distribution, provides clues about vertical transport, and is often utilized in dispersion studies (Weil, 1988, 1990). Further, Moeng and Rotunno (1990) find vertical velocity skewness is sensitive to the structure of the boundaries, *i.e.,* it depends on the type of surface boundary conditions, and also varies with Reynolds number in direct numerical simulations. Hunt et al. (1988) provides a brief interpretation of the skewness variation predicted by LES in the surface layer of a convective boundary layer.

The definition of the vertical velocity skewness is

$$S_w = \frac{\langle w^3 \rangle}{\langle w^2 \rangle^{3/2}} \qquad (6)$$

where $\langle \, \rangle$ denotes an ensemble average and w is the total velocity. In order to examine the impact of grid resolution on S_w we analyze the solutions from the different simulations in Table 1 with the caveat that we use the *resolved* or filtered vertical velocity $\overline{w}$. Hence we compute the resolved skewness

$$S_{\overline{w}} = \frac{\langle \overline{w}\,\overline{w}\,\overline{w} \rangle}{\langle \overline{w}\,\overline{w} \rangle^{3/2}}, \qquad (7)$$

from our LES solutions. Recall since typical LES uses Smagorinsky style closures with subgrid-scale fluxes parameterized at the second moment level subgrid-scale triple moments are unknown and thus there is not a clear definition of "subgrid-scale skewness" in an LES.

Vertical profiles of $S_{\overline{w}}$ are shown in figure 7. These profiles exhibit a clear and striking dependence on grid resolution; near the surface, $z/z_i < 0.15$, $S_{\overline{w}}$ decreases and eventually becomes (unrealistically) negative as the grid resolution decreases. Meanwhile as $z/z_i \to 1$ an opposite trend is observed. With decreasing grid resolution $S_{\overline{w}}$ becomes more positive and shows a pronounced maximum below the inversion. Away from the lower boundary, $0.05 < z/z_i < 1$, the skewness estimates appear to converge when the mesh is fine, 256^3 or greater. Notice the impact of grid resolution in the surface layer.
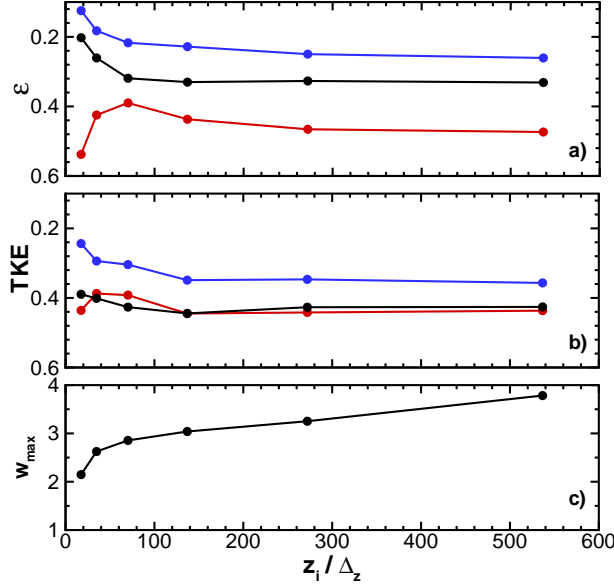
Figure 6: Effect of mesh resolution $z_i/\triangle z$ on bulk boundary layer turbulence. a) dissipation; b) total TKE; and c) the average resolved maximum vertical velocity. In a) and b) results for different vertical locations $z/z_i = (0.1, 0.5, 0.9)$ are indicated by (red,black,blue) curves, respectively.

As $z_i/\triangle z$ increases the skewness estimates, especially with meshes 512^3 and 1024^3, are in good agreement with the few available observations. Above $z/z_i > 0.75$, we have no compelling explanation for the differences between the fine mesh LES predictions and the few observations, but note that the presence of wind shear reduces the skewness (Fedorovich et al., 2001). There is an obvious need for more observations to determine whether this discrepancy is due to limited sampling in the observations or is a shortcoming of the LES.

The grid dependence in figure 7 invites further exploration. Some speculative explanations are: (1) grid resolution alters the structure of the overlying inversion. Coarser grids can only support weaker inversions compared to fine grids and perhaps $S_{\overline{w}}$ depends on inversion strength; or (2) perhaps the small scale high frequency content of $\langle \overline{w^3} \rangle$ changes sign below the inversion and thereby reduces the magnitude of $S_{\overline{w}}$ as the grid is refined.

Our current interpretation of the results in figure 7 hinges on the behavior and modeling of the subgrid-scale fluxes in LES. In order to expose this dependence we first introduce the definitions of the third and second order SGS moments

$$\phi = \overline{w^3} - \overline{w}^3 \equiv \overline{www} - \overline{w}\,\overline{w}\,\overline{w}, \qquad (8a)$$

$$\psi = \overline{w^2} - \overline{w}^2 \equiv \overline{ww} - \overline{w}\,\overline{w}. \qquad (8b)$$
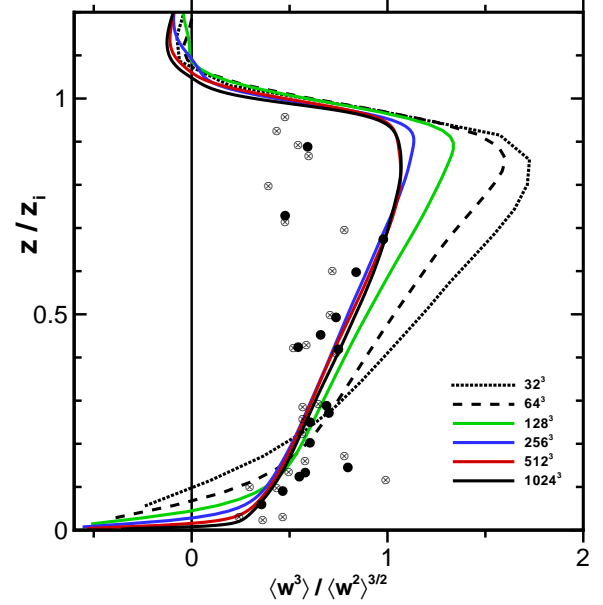

Figure 7: Effect of mesh resolution on vertical velocity skewness $S_{\overline{w}}$. The legend indicates the resolution of the various simulations. Note the skewness is computed using the resolved (or filtered) vertical velocity field $\overline{w}$. Observations are taken from the results provided in Moeng and Rotunno (1990).

As in usual LES practice $\overline{(\ )}$ indicates a spatially filtered variable in (8). Under the assumption that the filtering operator commutes with ensemble averaging, e.g.,

$$\langle w^3 \rangle \equiv \overline{\langle w^3 \rangle} = \left\langle \overline{w^3} \right\rangle, \qquad (9)$$

the total skewness given by (6) is next written in terms of resolved and subgrid contributions:

$$S_w = \frac{\langle \overline{w}^3 \rangle + \langle \phi \rangle}{(\langle \overline{w}^2 \rangle + \langle \psi \rangle)^{3/2}}. \qquad (10)$$

Further algebraic manipulation of (10) utilizing (8) leads to

$$S_w = S_{\overline{w}} \frac{(1 - \hat{\psi})^{3/2}}{(1 - \hat{\phi})} \qquad (11)$$

where $S_{\overline{w}}$ is the resolved-scale skewness (7) and

$$\hat{\phi} = \langle \phi \rangle / \left\langle \overline{w^3} \right\rangle, \qquad (12a)$$

$$\hat{\psi} = \langle \psi \rangle / \left\langle \overline{w^2} \right\rangle, \qquad (12b)$$

are non-dimensional SGS moments. (11) is useful – it defines the total skewness in terms of LES resolved and subgrid-scale variables. As might be expected, the subgrid contribution to the total skewness involves both second and third order moments of vertical velocity.
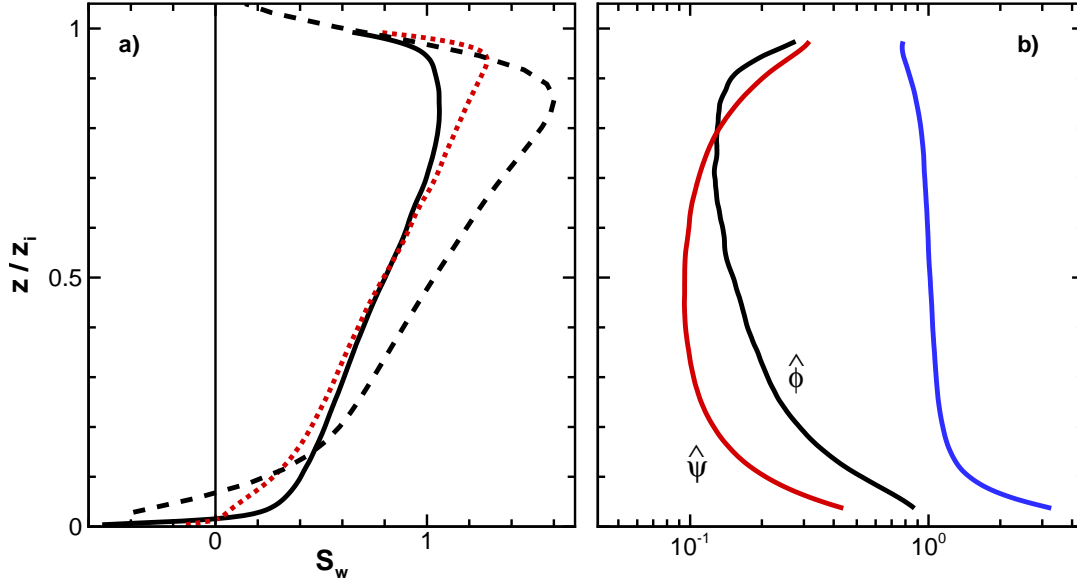
Figure 8: Panel a) skewness comparisons: 512^3 simulation, black line; 512^3 simulation filtered in horizontal planes to 64^2 resolution, red dotted line; and 64^3 simulation, black dashed line. Panel b) subgrid-scale moments computed from 512^3 simulation: $\hat{\phi}$, black line; $\hat{\psi}$, red line; and the SGS skewness correction $(1 - \hat{\psi})^{3/2}/(1 - \hat{\phi})$ [which appears in (11)], blue line.

In order to evaluate the importance of the SGS moments $(\hat{\phi}, \hat{\psi})$ to vertical velocity skewness we filtered the 512^3 and 1024^3 simulation results to produce resolved and SGS variables on a coarser mesh. This step is justified since the LES solutions, as shown previously, have effectively converged at these mesh resolutions. The vertical velocity field from cases E and F are filtered in horizontal $x - y$ planes to a resolution of 64^2 using a sharp spectral filter – no filtering is applied in the z direction. As an independent check on the processing we verified that the filtered fields satisfy (11) exactly.

Vertical profiles of skewness and SGS moments constructed from the filtered 512^3 simulation (referred to as case E_f) are presented in figure 8. Results obtained from filtering the 1024^3 simulation are similar but display more variability due to less averaging. The skewness estimates from E_f are intriguing. They are broadly similar to the comparable 64^3 coarse simulation result, *i.e.*, small in the surface layer and large near the inversion but exhibit important quantitative differences. In the surface layer the skewness from case E_f is always positive except very near the ground, in contrast to simulation B. This is in agreement with our physical expectation. Also the skewness from E_f matches the high resolution result in mid-PBL. The SGS moments in figure 8b illustrate the shortcomings of the coarse 64^3 calculation. In the surface layer the triple moment $\hat{\phi}$ is very large contributing more than 50% to $\langle \overline{w^3} \rangle$, in mid-PBL $\hat{\phi} \approx \hat{\psi}$, and near the

inversion $\hat{\phi} < \hat{\psi}$. $\hat{\phi}$ is always greater than zero. Overall the SGS "correction" to skewness given by the ratio on the right hand side of (11) is > 4 in the surface layer, ~ 1 in mid-PBL, and falls to ~ 0.8 near the inversion. We mention as $z \to z_i$ the strength of the PBL inversion might also alter the magnitude of the SGS moments and their relative contributions to S_w. As noted by Hunt et al. (1988), Smagorinsky closures are Gaussian models and hence assume $\hat{\phi} = 0$. As a consequence, coarse mesh LES results predict erroneous values of skewness because of their SGS closure schemes. In general, we find coarse mesh LES tends to overpredict $\langle \overline{w^3} \rangle$, underpredict $\langle \overline{w^2} \rangle$, and thus overpredict $S_{\overline{w}}$ compared to fine resolution simulations as shown in figure 9. When Smagorinsky closures are used with LES, meshes of at least 256^3 or greater are needed to obtain reliable estimates of $S_{\overline{w}}$. It will be interesting to examine vertical velocity skewness from LES with alternate non-eddy viscosity closure schemes, *e.g.*, Wyngaard (2004) and Hatlee and Wyngaard (2007) employ rate equations for the SGS fluxes and variances.

5. SUMMARY

A highly parallel LES code that utilizes 2-D domain decomposition and retains pseudospectral differencing in horizontal planes is described. The code exhibits good scaling over a wide range of problem sizes and is capa-
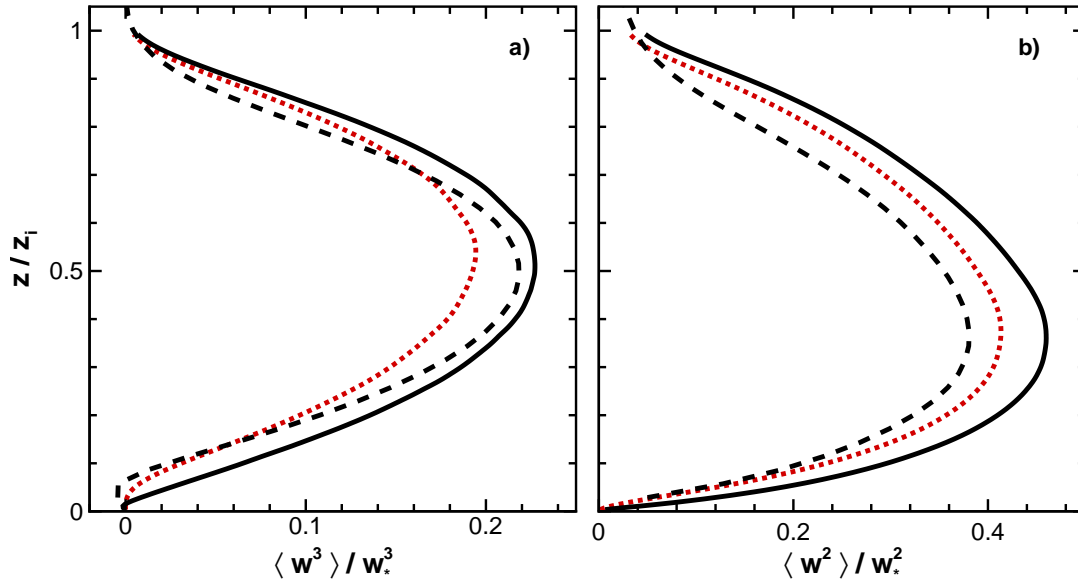
Figure 9: Comparison of third and second order vertical velocity moments from different calculations. 512^3 simulation, black line; 512^3 simulation filtered in horizontal planes to 64^2 resolution, red dotted line; and 64^3 simulation, black dashed line. Panel a) normalized $\langle \overline{w}^3 \rangle / w_*^3$ and panel b) normalized $\langle \overline{w}^2 \rangle / w_*^2$.

ble of using as many 16,384 processors of a Cray XT4. Flow visualization of fine mesh 512^3 and 1024^3 simulations of a convective boundary layer shows a number of intriguing structural features, *e.g.,* large scale plumes coupled to small scale vortical dust devils. A grid sensitivity study of a canonical daytime convective PBL shows that the LES solutions converge reasonably well for meshes greater than or equal to 256^3. The skewness of vertical velocity $S_{\overline{w}}$ highlights the solution sensitivity to grid resolution. The variations of $S_{\overline{w}}$ with grid resolution are a consequence of a Smagorinsky closure which neglects third-order SGS moments of vertical velocity. Future applications of this parallel code include high resolution simulations of air-wave-water interactions under high wind conditions and PBL couplings with surface-layer vegetation.

REFERENCES

Aoyama, Y. and J. Nakano, 1999: RS/6000 SP: Practical MPI programming, Technical Report IBM Redbook SG24-5380-00, International Business Machines.

Beare, R. J., M. K. Macvean, A. A. M. Holtslag, J. Cuxart, I. Esau, J.-C. Golaz, M. A. Jimenez, M. Khairoutdinov, B. Kosovic, D. Lewellen, T. S. Lund, J. K. Lundquist, A. McCabe, A. F. Moene, Y. Noh, S. Raasch, and P. P. Sullivan, 2006: An intercomparison of large-eddy simulations of the stable boundary layer, *Boundary-Layer Meteorol.*, **118**, 242–272.

Bretherton, C. S., M. K. Macvean, P. Bechtold, A. Chlond, W. R. Cotton, J. Cuxart, H. Cuijpers, M. Khairoutdinov, B. Kosovic, D. Lewellen, C.-H. Moeng, P. Siebesma, B. Stevens, D. E. Stevens, I. Sykes, and M. C. Wyant, 1999: An intercomparison of radiatively driven entrainment and turbulence

in a smoke cloud, as simulated by different numerical models, *Quart. J. Roy. Meteorol. Soc.*, **554**, 391–423.

Chandra, R., L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon, 2001: *Parallel Programming in OpenMP*, Academic Press, 230 pp.

Fedorovich, E., F. T. M. Nieuwstadt, and R. Kaiser, 2001: Numerical and laboratory study of a horizontally evolving convective boundary layer. Part i: Transition regimes and development of the mixed layer, *J. Atmos. Sci.*, **58**, 70–86.

Gibbs, W. R., 2004: A parallel/recursive algorithm, *J. Comp. Phys.*, **201**, 573–585.

Gropp, W., S. Huss-Lederman, A. Lumsdaine, E. Lusk, B. Nitzberg, W. Saphir, and M. Snir, 1998: *MPI-The Complete Reference,* Volume 2, The MPI-2 Extensions, The MIT Press, 344 pp.

Hatlee, S. C. and J. C. Wyngaard, 2007: Improved subfilter-scale models from the HATS field data, *J. Atmos. Sci.*, **64**, 1694–1705.

Hunt, J. C. R., J. C. Kaimal, and J. E. Gaynor, 1988: Eddy structure in the convective boundary layer - new measurements and new concepts, *Quart. J. Roy. Meteorol. Soc.*, **482**, 827–858.

Jonker, H. J. J., P. G. Duynkerke, and J. W. M. Cuijpers, 1999: Mesoscale fluctuations in scalars generated by boundary layer convection, *J. Atmos. Sci.*, **56**, 801–808.

Kanak, K. M., 2005: Numerical simulation of dust devil-scale vortices, *Quart. J. Roy. Meteorol. Soc.*, **131**, 1271–1292.

Lele, S. K., 1992: Compact finite difference schemes with spectral-like resolution, *J. Comp. Phys.*, **103**, 16–42.

Michalakes, J., J. Dudhia, D. Gill, T. Henderson, J. Klemp, W. Skamarock, and W. Wang, 2005: The Weather Research and Forecast Model: Software architecture and performance, in *Proceedings of the Eleventh ECMWF Workshop on the Use of High Performance Computing in Meteorology*, edited by W. Zwieflhofer and G. Mozdzynski, pp. 156–168, World Scientific.

Moeng, C.-H., 1984: A large-eddy simulation model for the study of planetary boundary-layer turbulence, *J. Atmos. Sci.*, **41**, 2052–2062.

Moeng, C.-H. and R. Rotunno, 1990: Vertical velocity skewness in the buoyancy-driven boundary layer, *J. Atmos. Sci.*, **47**, 1149–1162.

Moeng, C.-H. and P. P. Sullivan, 1994: A comparison of shear and buoyancy driven planetary-boundary-layer flows, *J. Atmos. Sci.*, **51**, 999–1022.

Moeng, C. H. and J. C. Wyngaard, 1988: Spectral analysis of large-eddy simulations of the convective boundary layer, *J. Atmos. Sci.*, **45**, 3573–3587.

—, 1989: Evaluation of turbulent transport and dissipation closures in second-order modeling, *J. Atmos. Sci.*, **46**, 2311–2330.

Nieuwstadt, F. T. M., P. J. Mason, C. H. Moeng, and U. Schumann, 1993: *Turbulent Shear Flows*, Springer-Verlag, 431 pp.

Patton, E. G., P. P. Sullivan, and C.-H. Moeng, 2005: The influence of idealized heterogeneity on wet and dry planetary boundary layers coupled to the land surface, *J. Atmos. Sci.*, **62**, 2078–2097.

Pekurovsky, D., P. K. Yeung, D. Donzis, W. Pfeiffer, and G. Chukkapallli, 2006: Scalability of a pseudospectral DNS turbulence code with 2D domain decomposition on Power4+/Federation and Blue Gene systems, in *ScicomP12 and SP-XXL*, Boulder,CO.

Raasch, S. and M. Schröter, 2001: Palm - A large-eddy simulation model performing on massively parallel computers, *Meteorologische Zeitschrift*, **10**, 363–372.

Spalart, P. R., R. D. Moser, and M. M. Rogers, 1991: Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions, *J. Comp. Phys.*, **96**, 297.

Sullivan, P. P., J. B. Edson, T. Hristov, and J. C. McWilliams, 2008: Large eddy simulations and observations of atmospheric marine boundary layers above non-equilibrium surface waves, *J. Atmos. Sci.*, **65**, 1225–1245.

Sullivan, P. P., T. W. Horst, D. H. Lenschow, C.-H. Moeng, and J. C. Weil, 2003: Structure of subfilter-scale fluxes in the atmospheric surface layer with application to large-eddy simulation modeling, *J. Fluid Mech.*, **482**, 101–139.

Sullivan, P. P., J. C. McWilliams, and W. K. Melville, 2007: Surface gravity wave effects in the oceanic boundary layer: Large-eddy simulation with vortex force and stochastic breakers, *J. Fluid Mech.*, **593**, 405–452.

Sullivan, P. P., J. C. McWilliams, and C.-H. Moeng, 1996: A grid nesting method for large-eddy simulation of planetary boundary layer flows, *Boundary-Layer Meteorol.*, **80**, 167–202.

Sullivan, P. P., C.-H. Moeng, B. Stevens, D. H. Lenschow, and S. D. Mayor, 1998: Structure of the entrainment zone capping the convective atmospheric boundary layer, *J. Atmos. Sci.*, **55**, 3042–3064.

UCAR/JOSS, 2005: Ad hoc committee and technical writing group for a petascale collaboratory for the geosciences, Establishing a Petascale Collaboratory for the Geosciences: Scientific Frontiers, Technical Report http://www.joss.ucar.edu/joss_psg/meetings/ Meetings_2005/petascale/index.html, UCAR/JOSS.

Weil, J. C., 1988: Dispersion in the convective boundary layer, in *Lectures on Air Pollution Modeling*, edited by A. Venkatram and J. Wyngaard, pp. 167–227, American Meteorological Society.

—, 1990: A diagnosis of the asymmetry in top-down and bottom-up diffusion in a Lagrangian stochastic model, *J. Atmos. Sci.*, **47**, 501–515.

Werne, J. and D. C. Fritts, 1999: Stratified shear turbulence: Evolution and statistics, *Geophys. Res. Lett.*, **26**, 439–442.

Wyngaard, J. C., 2004: Toward numerical modeling in the Terra Incognita, *J. Atmos. Sci.*, **61**, 1816–1826.