10 APPENDIX

PAPER:

S. Mukerji and J.M. McDonough, (1995), "Parallel Computation of 3-D Small-Scale Turbulence Via Additive Turbulent Decomposition", presented in *Parallel CFD'95*, California Institute of Technology, Pasadena, CA, June 1995.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

FG22-93PC93210

Parallel Computation of 3-D Small-Scale Turbulence Via Additive Turbulent Decomposition

S. Mukerji and J. M. McDonough*

Department of Mechanical Engineering
University of Kentucky, Lexington, KY 40506-0108, USA

Implementation and parallelization of additive turbulent decomposition is described for the small-scale incompressible Navier-Stokes equations in 3-D generalized coordinates applied to the problem of turbulent jet flow. It is shown that the method is capable of producing high-resolution local results, and that it exhibits a high degree of parallelizability. Results are presented for both distributed- and shared-memory architectures, and speedups are essentially linear with number of processors in both cases.

1. INTRODUCTION

Turbulent flows have been investigated both theoretically and experimentally for over a century, but in recent years the wide availability of supercomputers has spurred interest in numerical techniques. However most of the currently used computational methods have deficiencies and limitations. The drawbacks of modeling approaches like mixing length and κ-ϵ are well known, see for example [3]. Subgrid-scale modeling continues to be a weak part of the large eddy simulation (LES) technique (cf. Ferziger[1]), even when dynamic subgrid-scale models (Germano et al. [2]) are used. Direct numerical simulation (DNS) of turbulent flows is restricted to flows with low Reynolds numbers (Re) because of limitations of computing hardware; thus simulation of flow problems of engineering interest ($Re > 10^6$) is not presently feasible (see Reynolds [9]). Moreover, DNS has a limited scope for parallelization which means that it cannot fully utilize the opportunities offered by massively parallel processors (MPPs) which present the only hope of solving realistic turbulent flow problems in the near future.

The technique used for turbulent flow calculations in the present research is the additive turbulent decomposition (ATD) first proposed by McDonough et al. [7] and developed by McDonough and Bywater [4,5] in the context of Burgers' equation, and by Yang and McDonough [10] for the 2-D Navier-Stokes (N.-S.) equations. The algorithmic structure of ATD is similar to LES; but it uses unaveraged equations, and hence there is no closure problem. Like DNS, it maintains full consistency with the N.-S. equations. Moreover, ATD is designed to fully exploit the architecture of the MPPs and offers the possibility of both fine- and coarse-grained parallelization. The work presented here represents the first study of ATD in three space dimensions, and parallelization thereof. The results indicate

*This research is supported by the Department of Energy under Grant No. DE-FG22-93PC93210.

that theoretical speedups predicted in McDonough and Wang [6] and verified there with one-dimensional calculations are, in fact, achievable in three dimensions. In particular, it is shown in [6] that if linear speedups can be achieved on the three main parallelization levels of ATD (see Figure 2), then effective run times in turbulence simulations can be reduced to at most $\mathcal{O}(Re^{5/3})$ from the usual $\mathcal{O}(Re^3)$; our preliminary results indicate that required linear speedups are possible.

2. ADDITIVE TURBULENT DECOMPOSITION

2.1. Procedure

To demonstrate the details of the ATD algorithm it will be applied to the viscous, incompressible N.-S. equations which are given in vector form as:

$$\nabla \cdot \boldsymbol{U} = 0\,, \tag{1}$$

$$\boldsymbol{U}_t + \boldsymbol{U} \cdot \nabla \boldsymbol{U} = -\nabla P + \frac{1}{Re}\Delta \boldsymbol{U}\,. \tag{2}$$

The above equations are non-dimensional where $\boldsymbol{U}$ is the velocity vector scaled with respect to U_0, a characteristic velocity scale, and P is pressure non-dimensionalized by ρU_0^2 (ρ is density). Re is Reynolds number defined as $U_0 L/\nu$ where L is a length scale, and ν is kinematic viscosity. The gradient operator and the Laplacian are denoted by ∇ and Δ respectively. The subscript t denotes the time derivative in the transport equation.

The dependent variables are split into a *large-scale* $(\bar{\boldsymbol{U}}, \bar{P})$ and a *small-scale* $(\boldsymbol{U}^*, P^*)$ as follows:

$$\boldsymbol{U} = \bar{\boldsymbol{U}} + \boldsymbol{U}^* \quad \text{and} \quad P = \bar{P} + P^*\,.$$

The large-scale quantities can be viewed as the first few modes in a Fourier representation of the total quantity, and the small-scale quantities are the series remainders consisting of high mode numbers. The split quantities are then substituted into the governing equations which take the form:

$$\nabla \cdot (\bar{\boldsymbol{U}} + \boldsymbol{U}^*) = 0\,, \tag{3}$$

$$(\bar{\boldsymbol{U}}_t + \boldsymbol{U}^*_t) + (\bar{\boldsymbol{U}} + \boldsymbol{U}^*) \cdot \nabla(\bar{\boldsymbol{U}} + \boldsymbol{U}^*) = -\nabla(\bar{P} + P^*) + \frac{1}{Re}\Delta(\bar{\boldsymbol{U}} + \boldsymbol{U}^*)\,. \tag{4}$$

The governing equations are now additively decomposed (in a manner analogous to operator splitting schemes) into large-scale and small-scale equations:

$$\left.\begin{aligned} &\nabla \cdot \bar{\boldsymbol{U}} = 0\,, \\ &\bar{\boldsymbol{U}}_t + (\bar{\boldsymbol{U}} + \boldsymbol{U}^*) \cdot \nabla \bar{\boldsymbol{U}} = -\nabla \bar{P} + \frac{1}{Re}\Delta \bar{\boldsymbol{U}}\,, \end{aligned}\right. \qquad \text{(large-scale)} \tag{5}$$

$$\left.\begin{aligned} &\nabla \cdot \boldsymbol{U}^* = 0\,, \\ &\boldsymbol{U}^*_t + (\bar{\boldsymbol{U}} + \boldsymbol{U}^*) \cdot \nabla \boldsymbol{U}^* = -\nabla P^* + \frac{1}{Re}\Delta \boldsymbol{U}^*\,. \end{aligned}\right. \qquad \text{(small-scale)} \tag{6}$$

It is clear from equations (5-6) that there are enough equations for the number of unknowns; that is, there is no closure problem. The decomposition is not unique, but the

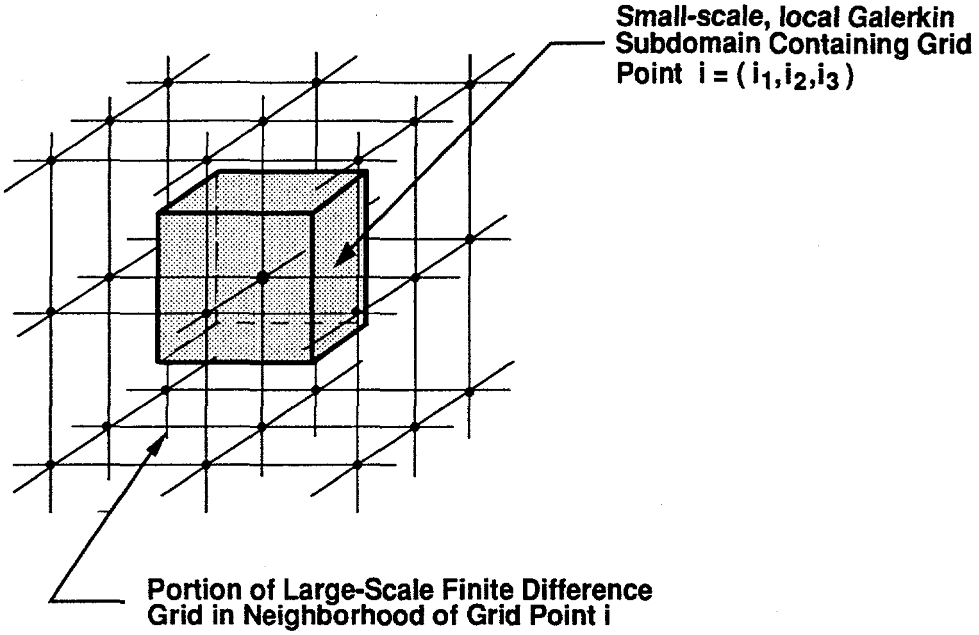
Small-scale, local Galerkin Subdomain Containing Grid Point i = (i1,i2,i3)
Portion of Large-Scale Finite Difference Grid in Neighborhood of Grid Point i

Figure 1. Domain decomposition employed in ATD

given form retains the structure of the N.-S. equations except for the additional *cross-terms*, e.g. $(\bar{U} \cdot \nabla U^*)$, which arises from decomposition of the nonlinear term of the original equation. The cross-terms also maintain coupling between the scales. Although at present the ATD algorithm is implemented with this two-level splitting, it could conceivably involve multi-level decomposition.

It should be noted that the consistency of the split equations with the N.-S. equations implies Galilean invariance. Also, realizability is automatically achieved because turbulent fluctuating quantities are calculated directly. The splitting also imparts the flexibility of using different numerical methods on each scale. Typically, finite difference (or finite volume) schemes are used for the large-scale which usually does not require high resolution, and Galerkin/spectral methods are used on the small-scale.

2.2. Parallelizability of ATD

Figure 1 shows a typical small-scale subdomain around a large-scale grid point. The small-scale equations are solved locally within this subdomain. The wide scope for parallelization in ATD is inherent in this spatial domain decomposition. Since there is a small-scale subdomain corresponding to each large-scale grid point, the small-scale solves can be parallelized easily.

Using a spectral method on the small scale converts the partial differential equations (PDEs) into a system of 1^{st}-order ordinary differential equations (ODEs). The evaluation of the right-hand side (RHS) of these ODEs at a given time level depends only on data from a previous time level. Hence within each small-scale solve the evaluation of the Galerkin ODE RHSs can be parallelized at each time step. Parallelization can be further implemented within each RHS evaluation to calculate the nonlinear convolutions and the cross-terms. Figure 2 shows these different levels of parallelization possible in ATD.

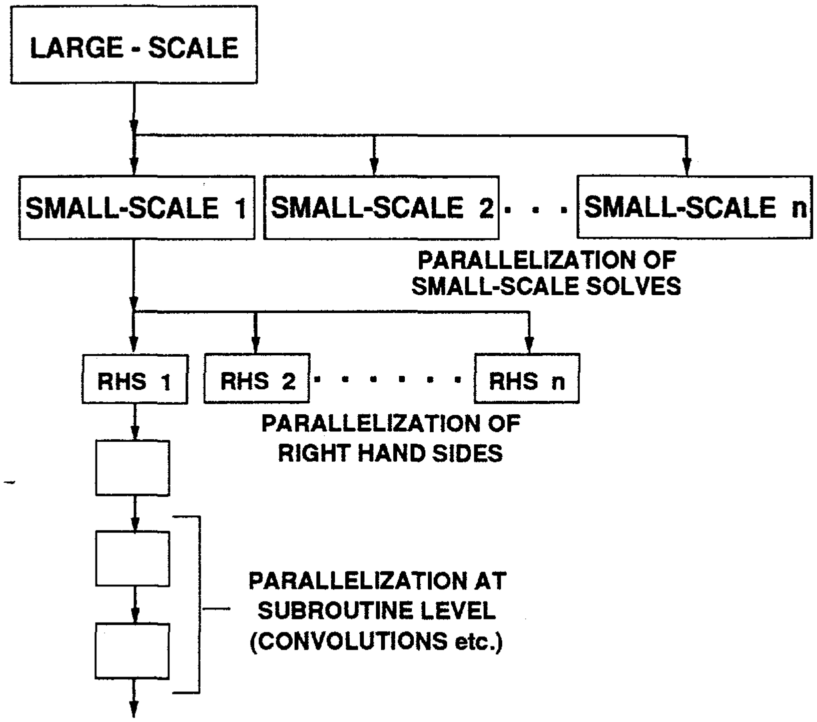
LARGE - SCALE
SMALL-SCALE 1
SMALL-SCALE 2
SMALL-SCALE n
PARALLELIZATION OF SMALL-SCALE SOLVES
RHS 1
RHS 2
RHS n
PARALLELIZATION OF RIGHT HAND SIDES
PARALLELIZATION AT SUBROUTINE LEVEL (CONVOLUTIONS etc.)

Figure 2. Levels of parallelization in ATD

3. PRESENT RESEARCH

At present the solution of only the small-scale equations is being attempted using parameterized inputs from the large-scale. This is a three-dimensional calculation in generalized coordinates. The large-scale is computed via a commercial CFD code.

The small-scale equations are solved using the Fourier-Galerkin spectral method described by Orszag [8]. In this technique the dependent variables are expressed as truncated triple Fourier series using complex exponential basis functions as

$$\boldsymbol{U}^*(\boldsymbol{\xi}, t) = \sum_{\boldsymbol{l}}^{\boldsymbol{N}} \boldsymbol{a}_{\boldsymbol{l}}(t) \exp(i\boldsymbol{\alpha}_{\boldsymbol{l}} \cdot \boldsymbol{\xi}) \,, \tag{7}$$

where $\boldsymbol{U}^*$ is now the small-scale contravariant velocity vector, $\boldsymbol{l}$ is the vector of mode indices, $\boldsymbol{N}$ represents the maximum number of Fourier modes in each direction, $\boldsymbol{\xi}$ is the position vector in generalized coordinates with origin at the large-scale grid point, $\boldsymbol{a}_{\boldsymbol{l}}(t)$ is the vector of time-dependent complex Fourier coefficients and $\boldsymbol{\alpha}_{\boldsymbol{l}}$ is the vector of wavenumbers which are scaled based on the large-scale grid spacing.

The Fourier representations of the dependent variables are then substituted into the governing equations, and the necessary operations are performed. Taking Galerkin inner products gives the spectral projection of the equations which are now a system of 1$^{\mathrm{st}}$-order ODEs in the time-dependent Fourier coefficients:

$$\frac{d\boldsymbol{a}_{\boldsymbol{l}}}{dt} = \boldsymbol{f}(\boldsymbol{a}_{\boldsymbol{l}}, \bar{U}) \,. \tag{8}$$

The function $\boldsymbol{f}$ is a notation to denote a rather complicated right-hand side vector. It should be noted that the Fourier coefficients of the small-scale solution depend explicitly on the large-scale solution $\bar{\boldsymbol{U}}$. This system of ODE initial value problems in the Fourier coefficients is integrated forward in time using Heun's method, an explicit 2nd-order Runge-Kutta scheme. Once the Fourier coefficients are known at a given time, the dependent variables can always be calculated using the Fourier series representation of equation (7).

4. RESULTS & DISCUSSION

4.1. Flow specifics & problem parameters

The particular flow problem under consideration is a three-dimensional turbulent jet. The flow is incompressible and the fluid has constant properties close to those of air: $\rho = 1.0$ kg/m^3 and $\mu = 1 \times 10^{-6}$ Ns/m^2. The jet diameter of 10 mm. is the reference length, and the total axial length of the domain is 760 mm. The jet velocity at the inlet is 5.0 m/s giving an inlet $Re = 5 \times 10^4$. The large-scale solution is obtained from a finite volume commercial CFD code on a rather coarse $13 \times 14 \times 81$ grid.

Figure 3 displays an instantaneous spatial slice of the large-scale solution (velocity vectors and pressure) and the location of the small-scale subdomain. The subdomain has a small radial offset, and its axial position is halfway between the inlet and outlet. This gives a non-dimensional axial location of 38, which is well beyond the potential flow core and in the region of fully-developed turbulence. Adequate resolution on the small-scale was obtained by $\boldsymbol{N} = (7, 7, 7)$ Fourier modes. The choice of the small-scale time step was dictated by the stiffness of the problem and the explicit nature of the solution method; it was set to 2×10^{-6} sec. to maintain stability.

4.2. Turbulence calculations

The main motivation for the present research is to obtain highly resolved small-scale turbulent solutions. For this purpose the code was allowed to perform time integrations for relatively long times corresponding, approximately, to the large-scale time scale. Figure 4 shows the time series of the circumferential component of small-scale velocity at a

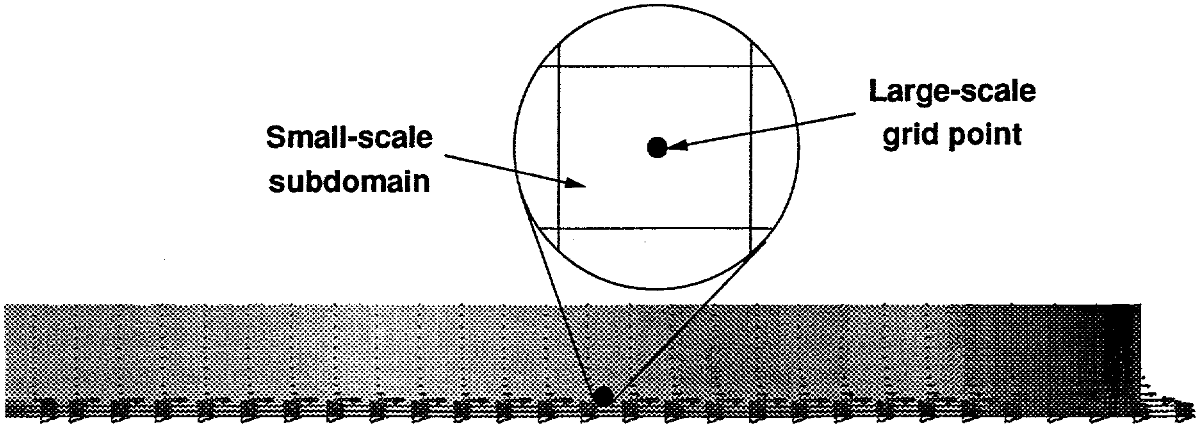

Figure 3. Large-scale flowfield and location of small-scale domain

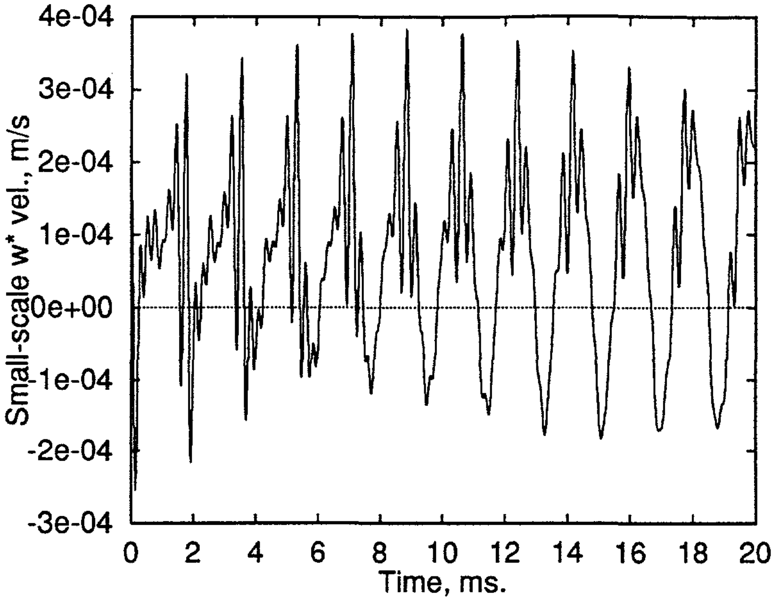

Figure 4. Time series of small-scale velocity

location in the center of the small-scale subdomain. The time series shows strong residual periodicity with a time scale of about 2 ms. This is probably to be expected in a region where the jet profile is self-similar. Moreover, these fluctuations are truly small scale having magnitudes of order 10^{-4} compared with $\mathcal{O}(1)$ for the large-scale velocity. This demonstrates the high degree of spatial and temporal resolution achievable with ATD.

4.3. Parallelization results

A broad, coarse-grained parallelization of the ODE right-hand side evaluations was implemented. The code was allowed to integrate the solution forward in time for fifty small-scale time steps. Figure 5 shows the speedups obtained on a distributed-memory Convex-HP MetaSystem consisting of 8 processors running PVM. Parallelization was also implemented using compiler directives on a 24 processor shared-memory Convex-HP Exemplar SPP-1000. The corresponding speedup results are shown in Figure 6.

The data partitioning used to divide the computational load among different processors is very effective for both machines. This is evident from the balanced CPU load on the parallel threads as shown in Figure 7. The CPU time required to solve the problem using 8 processors was about 300 sec. on the MetaSystem and about 120 sec. on the Exemplar. Thus the code runs much faster under the shared-memory paradigm, and this is despite the fact that the processors on the Exemplar currently have a slightly lower clock speed.

It can be seen that the speedups achieved with multiple processors scale essentially linearly with the number of processors. It should be recalled from Figure 2 that parallelization can be implemented at levels both above and below the present one. This implies that with the availability of true MPPs ($\mathcal{O}(10^3)$ processors, or more) the ATD algorithm can perform full three-dimensional turbulence simulations in physically realistic situations involving generalized coordinates, at *Re* at least as high as 10^5.

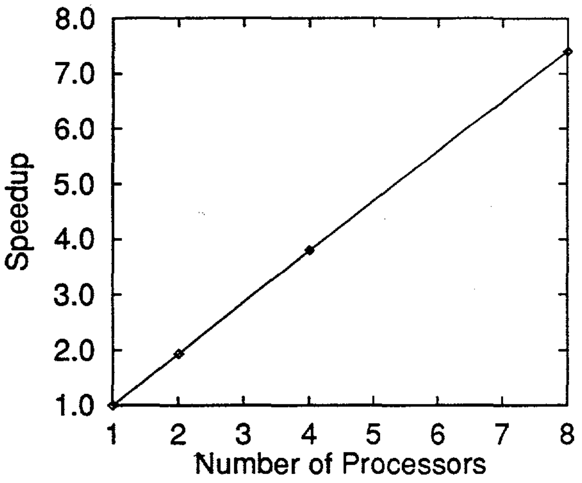

Figure 5. Parallelization speedup on Convex-HP MetaSystem

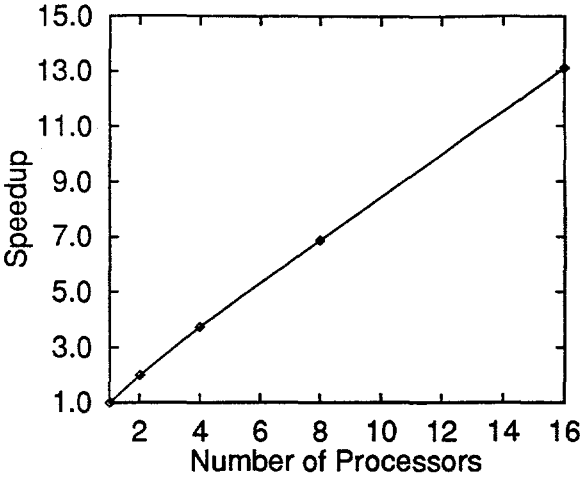

Figure 6. Parallelization speedup on Convex-HP Exemplar

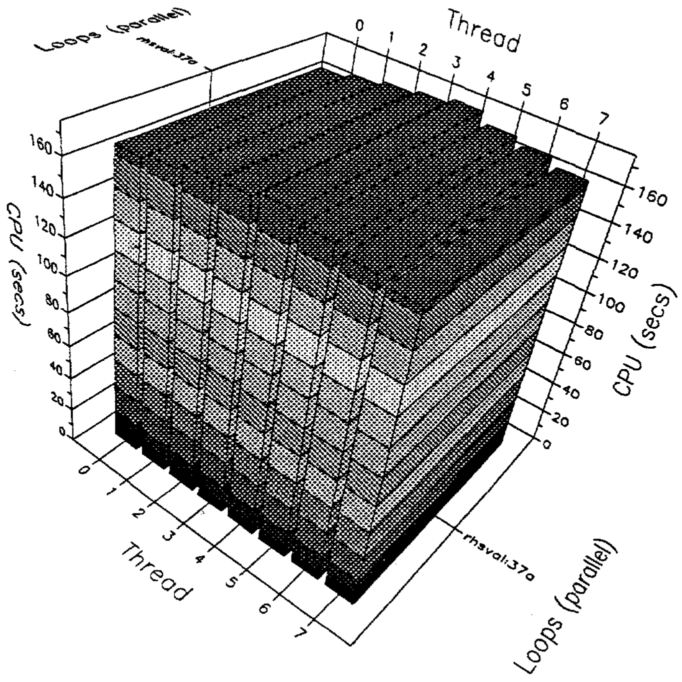

Figure 7. CPU load balance on parallel threads (Profile generated for 8 processors on Convex-HP Exemplar)

5. SUMMARY & CONCLUSIONS

This is the first implementation of the ATD algorithm in three space dimensions. The two main (theoretically expected) features of ATD are its parallelizability and its ability to produce highly resolved solutions; both are demonstrated in the present results.

Parallelization in ATD can be implemented at several different levels which can be nested. Parallelization speedups scale essentially linearly with number of processors for an intermediate (and thus more difficult to parallelize) level. This implies that high *Re* three-dimensional turbulence simulations are possible on MPPs.

The present code parallelizes well under both shared-memory and distributed-memory paradigms. Programming is much easier under shared memory using a few compiler directives than under distributed memory which requires learning a message passing language. The execution times are also much less under shared memory, even with somewhat slower processors. We conclude from this that, despite a rather widespread early fear of shared-memory architectures, they can perform at least as well as distributed-memory machines. We believe this makes them the obvious choice for future MPP architectures due to their ease of use.

REFERENCES

1. J. H. Ferziger, "Higher-level Simulations of Turbulent Flows," in *Computational Methods for Turbulent, Transonic and Viscous Flows*, Essers (ed.), Hemisphere Pub. Corp., Washington DC, 1983.
2. M. Germano, K. Piomelli, P. Moin and W. H. Cabot, "A Dynamic Subgrid-scale Eddy Viscosity Model," *Phys. Fluids* A3, 1760, 1991.
3. J. M. McDonough, "On the effects of modeling errors in turbulence closures for the Reynolds-averaged Navier-Stokes equations," Report CFD-03-93, Department of Mechanical Engineering, University of Kentucky, 1993.
4. J. M. McDonough and R. J. Bywater, "Large-Scale Effects on Small-Scale Chaotic Solutions to Burgers' Equation," *AIAA J.* 24, 1924, 1986.
5. J. M. McDonough and R. J. Bywater, "Turbulent Solutions from an Unaveraged, Additive Decomposition of Burgers' Equation," in *Forum on Turbulent Flows-1989*, Bower & Morris (eds.), FED Vol. 76, ASME, NY, 1989.
6. J. M. McDonough and D. Wang, "Additive turbulent decomposition: A highly parallelizable turbulence simulation technique," Report CFD-02-94, Department of Mechanical Engineering, University of Kentucky, 1994. Submitted to *Int. J. Supercomput. Appl. High Perform. Comput.*.
7. J. M. McDonough, J. C. Buell and R. J. Bywater, "An Investigation of Strange Attractor Theory and Small-Scale Turbulence," *AIAA Paper 84-1674*, 1984.
8. S. A. Orszag, "Numerical Methods for the Simulation of Turbulence," *Phys. Fluids, Supplement II*, 12, 250, 1969.
9. W. C. Reynolds, "The Potential and Limitations of Direct and Large Eddy Simulation," in *Whither Turbulence? Turbulence at the Crossroads*, Lumley (ed.), Springer-Verlag, Berlin, 313, 1990.
10. Y. Yang and J. M. McDonough, "Bifurcation Studies of the Navier-Stokes Equations Via Additive Turbulent Decomposition," in *Bifurcation Phenomena and Chaos in Thermal Convection*, Bau et al. (eds.), HTD Vol. 214, ASME, NY, 1992.