HIGH PERFORMANCE PSEUDO-SPECTRAL FLUID FLOW COMPUTATIONS AT NACAD/COPPE/UFRJ

N. Mangiavacchi, A. L. G. A. Coutinho, N. F. F. Ebecken

NACAD - COPPE/UFRJ
PO Box 68516, 21945-970, Rio de Janeiro, RJ
Phone: (21) 290 7116, E-mail: norberto@coc.ufrj.br

Resumo

Usando a mesma estratégia numérica e de paralelização, métodos de alto desempenho baseados em métodos pseudo-espectrais para a simulação de escoamentos turbulentos, escoamentos bidimensionais bifásicos, e escoamentos em meios porosos foram desenvolvidos. Os programas foram desenvolvidos com o proposito de realizar simulações com alta resolução, com precisão espacial e temporal de escoamentos complexos, em computadores de varias arquitecturas, inclusive computadores de memória distribuida. Os programas oferecem bons "speedup" em máquinas paralelas, e bom desempenho em máquinas vetoriais.

Abstract

Using the same numerical and parallelization strategy, high performance pseudo-spectral methods for the simulation of turbulent flows, two-dimensional two-phase flows, and flows in porous media were developed. The codes are designed to perform high resolution, space and time accurate simulations of complex flows on various computer architectures, including distributed memory systems. The codes give good speedups on parallel machines, and good performance on vector machines.

1 Introduction

The numerical simulation of complex flows is of great importance in many engineering and scientific applications. Many of these applications deal with turbulent flows, flows in porous media, multiphase flows. Numerical simulation of these complex flows require the intensive utilization of computational resources. In this work we describe the application of pseudo-spectral methods to the solution of complex flow problems in high performance vector and parallel computers. The pseudo-spectral codes were designed to perform high resolution, space and time accurate simulation of turbulent flows, two-phase flows, and flows in porous media on various current architectures, including distributed memory architectures. The parallel algorithms explore the intrinsic parallelism of the pseudo-spectral methods and are based on a domain decomposition approach.

2 The numerical methods

The same numerical and implementation strategy was used in the development of the various flow solvers. The governing equations for incompressible turbulent flows [1], two-phase bubbly flows [2,3], and flows in porous media [4,5] have been presented elsewhere. They consist of sets of coupled partial differential equations that include both elliptic and parabolic behavior. We use a Fourier Pseudo-Spectral Method in order to discretize the system of partial differential equations into ordinary differential equations. The pseudo-spectral approach avoids the high cost of computing quadratic and higher terms in Fourier representation, which require convolutions, but still preserving the *Spectral Convergence* properties of standard spectral methods. The Pseudo-spectral approach gives the best computational cost/benefit for simple geometries, in particular for periodic domains. We proceed using the expansion of the flow variables in Fourier series in the coordinate x, y, and z directions,

$$\mathbf{u}(\mathbf{x}) = \sum_{\mathbf{k}} \mathbf{U}(\mathbf{k}) \exp(ik_x x) \exp(ik_y y) \exp(ik_z z) \tag{1}$$

The time integration for the transport equation is performed using operator splitting. The Leap-frog scheme and Adams-Bashfort are used for the advection terms of the scalar transport, and for the advection in the momentum equation, while the fully implicit scheme is used for the viscous and diffusion terms. In the case of the porous media solver [5], the vorticity is computed from the concentration field, and the velocities are computed directly solving two Poisson operators, whithout solving explicitly for the pressure. For the turbulent flow and bubbly flow solvers, the

of two-phase bubbly flows [3], the tracking of the position of the interface between the two phases is done using the transport equation for one of the phases, and the computation of surface tension is done using the gradient of concentration of one of the phases. The program for incompressible turbulent flow will be used for the development of new subgrid-scale models for turbulent flows in atmospheric and industrial applications. The program for two-phase bubbly flow will be used for the study of the effect of the presence of particles of an insoluble phase in the dynamics of turbulent flows, and for the study of turbulence generation in two-phase flows. The program can also be applied in the study of the collision and coalescence of drops in turbulent flows. The program for flow in porous media will be used to study the hydrodynamic instability of miscible displacements.

3 Parallel algorithm

The parallel algorithm explores intrinsic parallelism of the pseudo-spectral technique. It is based on a domain decomposition approach in *latu sensu*. Three kinds of operations are involved in the parallel pseudo-spectral method: (1) computation of products in physical space, (2) inversion of the Poisson operators, computation of derivatives, and filtering operations in Fourier space, and (3) computation of the discrete two-dimensional Fourier transforms. The first two kinds of operations can be performed without any communication, when the domain is partitioned among the processors, in physical space and in Fourier space respectively. The only part that requires communication is the computation of the the discrete two-dimensional Fourier transforms. Here also, for portability, we have chosen to use the transposition approach, that allows to use highly optimized single processor one-dimensional FFT routines, that are normally found in most architectures, and a transposition algorithm that can be easily ported to different distributed memory architectures. The transposition algorithm used in the FFT is the so called *direct* transposition algorithm where at each stage of the communication algorithm each node sends to his pair all the data that has that node as final destination. At each stage the processor pairs are defined using a mapping of the processors onto a hypercube, and a relative addressing strategy. The number of stages is $p-1$ where p is the number of processors. The transposition algorithm can be summarized as follows:

- For i=1, 2, ...,$p-1$ do:
  - Each node m collects and sends to $n = XOR(m,i)$ all blocks that have node n as final destination, and replaces them with the blocks that re-

- Unshuffle the resulting data in each node.

The Poisson operator is diagonal in Fourier space. After decomposition of the domain among the nodes, the Poisson operator will require $O(\frac{N^d}{p})$ computations per node, at each time step, and no communications. Here N is the resolution in each dimension, and d is the number of dimensions. The multidimensional FFT's will require $O(\frac{dN^d \log_2(N)}{p})$ computations and $(p-1)$ bidirectional communications of length $\frac{N^d}{p^2}$ when using the described transposition algorithm. Since the latency time is much shorter than the communication time for large problem sizes, the total communication time is essentially $O(\frac{N^d}{p})$. Therefore the ratio of communications to computations is $O(\frac{1}{\log_2(N)})$.
Except for a few global operations, all the communication are lumped in the transposition algorithm making it easily portable to other distributed memory computers. The implementation using PVM, as the one given in [5], can run on a CRAY T3D and an IBM SP2 without changes in the transposition algorithms, only requiring to load the appropriate machine dependent single-processor one-dimensional FFT routines. To achieve better performance on the SP2, versions employing the MPI and MPL libraries were also implemented with minor additional effort, and considerable improvement in the performance.

4 Single node and parallel performance

To evaluate the performance of the code in parallel machines, two-dimensional computations were performed with resolutions ranging from 256×256 to 1024×1024.

Table 1: Single-processor performance (256×256 resolution), flow in porous media.

CPU	FFT	TIME(s/100 iters)	*Mflop/s*
IBM SP2 wide	Fortran	246	10
IBM SP2 thin	ESSL	90	28
IBM SP2 wide	ESSL	49	50
CRAY T3D	scalar library	156	16
CRAY J90	vector library	24.7	100
CRAY T90	vector library	3.38	731

High single-processor performance is obtained across various platforms using the provided optimized high performance libraries for the FFTs. The first and third

single-processor performance in the IBM SP2 can be improved by a factor of 5 by using the ESSL library. On CRAY T3D implementation, the scalar FFT library subroutines CCFFT,SCFFT, and CSFFT were used. On the CRAY J90 and T90 version, the vector library FFT subroutines CFFTMLT, SCFFTM, and CSFFTM where used.

Parallel performance was measured on a 4-processor IBM SP2. and on a 32-processor CRAY T3D. When using the PVM libraries on the SP2, the speed-up curves show that the parallel efficiency drops to about 75% for two processors. This is caused by the transposition algorithm, which is not present in the single processor case, and that introduces a significant overhead even when using only 2 processors. However efficiency continues above 50% even for quite larger numbers of nodes, as long as the problem size is adequate. When using the dedicated MPL library on the SP2 we obtain some improvement in the efficiency (on two processors about 80% for 256×256), showing that the communication cost is reduced by using the MPL library.

In fact, the efficiency is about 50% even when using 32 T3D nodes for the case 1024×1024. Hence, a CRAY T3D with 32 nodes outperforms a single CPU of a CRAY J90 running in vector mode. The scalability of the parallel code clearly indicates its potential to beat more powerful vector processors, as the T90, if more processors were added to the parallel machine.

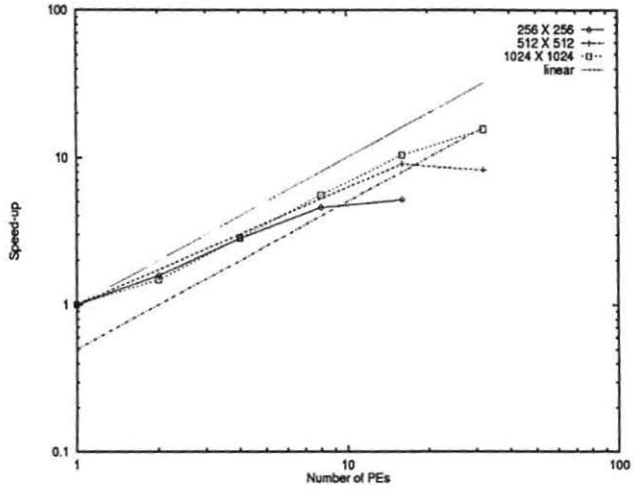

Figure 1: Speedup on CRAY T3D. Flow in porous media.

5 Conclusions

Using the same numerical and parallelization strategy, high performance parallel pseudo-spectral methods for the simulation of turbulent flows, two-dimensional two-phase flows, and flows in porous media were developed. The codes are designed to perform high resolution, space and time accurate simulations of complex flows on various distributed memory architectures.

The parallel Pseudo-Spectral codes are efficient on various architectures. They give good speedup on distributed memory systems (IBM SP2, and CRAY T3D), and good performance on vector machines (CRAY J90 and T90).

Among the future improvements that will be added to the code are the implementation of parallel asynchronous communications, to partially hide the communication cost of the transposition.

Acknowledgements

We gratefully acknowledge the Center of Parallel Computations of the Federal University of Rio de Janeiro (NACAD-COPPE) for providing us time on the IBM SP2, and CRAY J90 machines. We are also indebted to Silicon Graphics/Cray Research Division by the computer time in their CRAY T3D and T90 machines at Eagan, MN, USA. The first author is supported by a Post-doctoral Research Grant from the Brazilian Research Council (CNPq 150013/96-5 NV).

References

1. Mangiavacchi, N., Akhavan, R., Direct numerical simulations of turbulent shear flows on distributed memory architectures, Proc. 6th SIAM Conference on Parallel Processing for Scientific Computing, Vol 1, pp. 61-4, 1993.

2. Esmaeeli, A. and Tryggvason, G., An Inverse Energy Cascade in Two-dimensional Low Reynolds Number Bubbly Flows, J. Fluid Mech., Vol. 314, pp. 315-330, 1996.

3. Mangiavacchi, N, Coutinho, A.L.G.A., Ebecken, N.F.F., Parallel pseudo-spectral method for two-phase flows, to appear on Proc. XVIII Congresso Ibero Latino-Americano de Metodos Computacionais em Engenharia, Brasilia, 1997.

4. Zimmerman, W. B., Homsy, G. M., Nonlinear viscous fingering in miscible displacements with anisotropic dispersion, Phys. Fluids, **A** 3 (8), pp. 1859-72, 1991.

5. Mangiavacchi, N, Coutinho, A.L.G.A., Ebecken, N.F.F., Parallel pseudo-spectral simulations of nonlinear viscous fingering in miscible displacements, to appear on Proc. IX International Symposium on Offshore Engineering, Rio de Janeiro, 1997.