8th. World Congress on Computational Mechanics (WCCM8)
5th European Congress on Computational Methods in Applied Sciences and Engineeering (ECCOMAS 2008)
June 30 –July 5, 2008
Venice, Italy

Strong linear scaling for spectral simulations of time dependent semilinear partial differential equations on Marenostrum

R. de la Cruz[1], *B.K. Muite[2] and H. Servat[1]

[1]Barcelona Supercomputing Center,
C/ Jordi Girona, 29,
08034 Barcelona, Spain
delacruz@bsc.es
harald.servat@bsc.es

[2]Mathematical Institute,
University of Oxford,
22-26 St. Giles, Oxford,
OX1 3LB, UK
muite@maths.ox.ac.uk

Key Words: *Spectral methods, Distributed memory architectures, Fast Fourier Transform.*

ABSTRACT

We solve a time dependent semilinear partial differential equation using a spectral collocation method on a distributed memory supercomputer. Previous attempts to use spectral methods to solve evolutionary partial differential equations have scaled poorly on distributed memory machines because typical time stepping algorithms require fast global all-to-all communications. Consequently, primarily expensive supercomputers with very fast interprocessor communications are used to do large scale spectral simulations – see for example [1]. More common distributed memory machines do not have the fastest possible interprocessor communications. By using a modified finite difference time stepping scheme, we overlap communication and computation to obtain almost linear strong scaling for upto 512 processors on a two dimensional scalar partial differential equation discretized using 8192×8193 grid points.

The partial differential equation we solved numerically is

$$\rho u_{tt} - \beta \Delta u_t = (u_x^3 - u_x)_x + u_{yy} - \epsilon^2 \Delta^2 u,$$

where ρ is the material density, u is the displacement, t is time, β is the viscosity, x is the original position in the reference configuration and ϵ is the capillarity. Mathematical results and related references on this viscoelastic model can be found in [2]. Space is discretized using Fourier modes in the x direction and Chebyshev modes in the y direction – see [3] for an introduction to spectral methods. The second order finite difference time stepping scheme is

$$\begin{aligned} &\rho \left(2u^{n+1} - 5u^n + 4u^{n-1} - u^{n-2}\right) / \left(\delta t^2\right) - \beta \left(3u_{xx}^{n+1} - 4u_{xx}^n + u_{xx}^{n-1}\right) / \left(2\delta t\right) \\ &= 4(u_x^{n-2})_x^3 - 3(u_x^{n-3})_x^3 - u_{xx}^{n+1} + u_{yy}^{n+1} - \epsilon^2 \Delta^2 u^{n+1} \end{aligned}$$

where the superscript n denotes the time step. Time stepping takes place in spectral space, but to obtain the nonlinear term, $(u_x^{n-2})_x^3$, u_x^{n-2} is transformed to real space where the multiplication $(u_x^{n-2})^3$ is performed, after which $(u_x^{n-2})^3$ is transformed back to spectral space and differentiated.

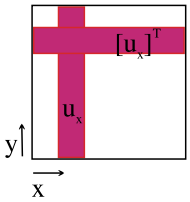

Figure 1: Shaded area shows data for u_x held on one of many processors.

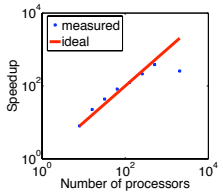

Figure 2: Comparison of ideal speedup with measured speedup.

On a single processor the new solution, u^{n+1}, is obtained using a loop over the linearly uncoupled Fourier modes in which a linear Chebyshev system is solved. Once a solution for each Fourier mode is obtained, it is transformed out of Chebyshev space using a one dimensional fast Chebyshev transform. A loop over the Chebyshev modes is then done where the x derivative of the solution, u_x^{n+1}, is transformed from Fourier space to real space, the nonlinear term, $(u_x^{n+1})^3$, is computed transformed back to Fourier space and then differentiated. Then a loop over the x direction is initiated in which the appropriate combination of stored nonlinear terms, $4(u_x^{n-1})_x^3 - 3(u_x^{n-2})_x^3$, is transformed to Chebyshev space and the solution at the next time step is computed.

When this scheme is parallelized, one dimensional fast transforms are still done on single processors, thus, as shown in Fig. 1, each processor holds a portion of the modes of u_x and of $[u_x]^T$, so that Chebyshev transforms are done on u_x and Fourier transforms are done on $[u_x]^T$. In standard time stepping schemes, u^{n+1} is computed using an approximation to the nonlinear term, $(u_x^3)_x$, that depends on u_x^n, and so fast interprocessor communications are required to transpose u_x^n and obtain the nonlinear term $(u_x^n)_x^3$. The time-stepping scheme used here hides latency, because the nonlinear term is extrapolated using u_x^{n-2} and u_x^{n-3}. In this parallelized scheme, u^{n+1} is computed while simultaneously receiving the nonlinear term $4(u_x^{n-1})_x^3 - 3(u_x^{n-2})_x^3$. The nonlinear term $\left[4(u_x^n)_x^3 - 3(u_x^{n-1})_x^3\right]^T$ is then calculated while $\left[u_x^{n+1}\right]^T$ is received. In the next iterate, u^{n+2} is computed using $4(u_x^{n-1})_x^3 - 3(u_x^{n-2})_x^3$ while $4(u_x^n)_x^3 - 3(u_x^{n-1})_x^3$ is received.

Figure 2 shows strong scaling for a discretization with 8192×8193 grid points. A discussion of computational efficiency and numerical solutions of geometrically nonlinear vectorial viscoelastic models for martensitic phase transformations will be presented at the conference.

ACKNOWLEDGEMENTS

This work was carried out under the HPC-EUROPA project (RII3-CT-2003-506079), with the support of the European Community - Research Infrastructure Action under the FP6 “Structuring the European Research Area” Program.

REFERENCES

[1] M. Yokokawa, K. Itakura, A. Uno, T. Ishihara, and Y. Kaneda. “16.4 Tflops direct numerical simulation of turbulence by a Fourier spectral method on the Earth Simulator”. *SC2002: From Terabytes to Insights*, 2002.

[2] K.-H. Hoffman and P. Rybka. “On convergence of solutions to the equation of viscoelasticity with capillarity”. *Comm. Partial Differential Equations*, Vol. **25**, 1845–1890, 2000.

[3] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Fundamentals in Single Domains*, Springer, 2006.