JOURNAL OF COMPUTATIONAL PHYSICS 105, 371 (1993)

Abstracts of Papers to Appear in Future Issues

EFFICIENT PARALLEL IMPLEMENTATION OF MOLECULAR DYNAMICS ON A TOROIDAL NETWORK. PART I. PARALLELIZING STRATEGY. Klaas Esselink, Berend Smit, and Peter Hilbers. *Koninklijke/Shell-Laboratorium, Amsterdam, Shell Research B.V., Badhuisweg 3, 1031 CM Amsterdam, The Netherlands.*

Molecular dynamics simulations require supercomputers. A specific class of supercomputers is that of parallel computers. We derive an implementation of molecular dynamics on a toroidal network of processors. First, we argue that for a fast algorithm the simulation universe has to be divided into regular cells, and we determine the best shape of these cells. For a parallel implementation, we choose to distribute cells rather than particles and we show how to assign the cells to processors, given certain restrictions on universe and network. The assignment is proven to be optimal with respect to communication cost. We go on to explain our implementation. Finally, we compare the timing results with those for computations performed on a Cray single-processor machine. The physical results obtained with the implementation are discussed elsewhere.

EFFICIENT PARALLEL IMPLEMENTATION OF MOLECULAR DYNAMICS ON A TOROIDAL NETWORK. PART II. MULTI-PARTICLE POTENTIALS. K. Esselink and P. A. J. Hilbers. *Koninklijke/Shell-Laboratorium, Amsterdam, Badhuisweg 3, 1031 CM Amsterdam, The Netherlands.*

Implementations for molecular dynamics on parallel computers generally use either particle parallelism or geometric parallelism. For short-range potentials, geometric parallelism has the advantage that communication can stay restricted to processors nearby. Usually, half the environment around a processor is communicated, using Newton's third law. This poses a problem for the implementation of multi-particle potentials (e.g., "bending" and "torsion"). For instance, if it is said that only one processor should actually calculate the forces on the particles involved, it will be difficult to determine which processor this should be, given that the particles are distributed over two or more processors. We present an efficient technique to do so and prove that it is correct. The technique requires no more communication than the computation of two-particle interactions and ensures that potentials are only evaluated once.

WAVELETS AND THE NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS. Sam Qian and John Weiss. *Aware, Inc., One Memorial Drive, Cambridge, Massachusetts 02142, USA.*

We present a new numerical method for the solution of partial differential equations in nonseparable domains. The method uses a wavelet-Galerkin solver with a nontrivial adaptation of the standard capacitance matrix method. The numerical solutions exhibit spectral convergence with regard to the order of the compactly supported, Daubechies wavelet basis. Furthermore, the rate of convergence is found to be independent of the geometry. We solve the Helmholtz equation since, for variations in the parameter, the solutions have qualitative properties that well illustrate the applications of our method.

CHEBYSHEV COLLOCATION METHOD AND MULTI-DOMAIN DECOMPOSITION FOR NAVIER–STOKES EQUATIONS IN COMPLEX CURVED GEOMETRIES. C. R. Schneidesch and M. O. Deville. *Université Catholique de Louvain, Unité de Mécanique Appliquée, Louvain-La-Neuve, Belgium.*

A general multidomain decomposition is proposed for the numerical solution of the 2D incompressible stationary Navier–Stokes equations. The solution technique consists in a Chebyshev orthogonal collocation method preconditioned by a standard Galerkin finite element technique. The preconditioned system is then solved through a Richardson procedure. The domain of interest is decomposed into quadrilaterals, curved when needed. A Gordon transfinite interpolation performs the curvilinear grid generation of the obtained simply-connected planar subdomains. The interface conditions, naturally incorporated into the finite element approach, relate neighbour subdomains through the normal jump of appropriate fluxes across internal boundaries, where an integral form of C^1 continuity is consequently achieved at convergence of the iterative processes. The study of model Stokes problems demonstrates that the current method still behaves spectrally in distorted geometries. For curvilinear distortion, a loss of several orders of magnitude is observed in the solution accuracy even when the distortion is very limited. Finally, some results of flow simulation in a constricted channel are proposed to illustrate the abilities of the present method to treat Navier–Stokes problems.

A FAST RECURSIVE ALGORITHM FOR MOLECULAR DYNAMICS SIMULATION. A. Jain. *Jet Propulsion Laboratory/California Institute of Technology, 4800 Oak Grove Drive, Pasadena, California 91109, USA;* N. Vaidehi. *A. A. Noyes Laboratory of Chemical Physics, California Institute of Technology, Pasadena, California 91125, USA;* G. Rodriguez. *Jet Propulsion Laboratory/California Institute of Technology, 4800 Oak Grove Drive, Pasadena, California 91109, USA.*

In this paper, we develop a recursive algorithm for solving the dynamical equations of motion for molecular systems. We make use of internal variable models which have been shown to reduce the computation times of molecular dynamics simulations by an order of magnitude when compared with Cartesian models. The $O(\mathcal{N})$ algorithm described in this paper for solving the equations of motion provides additional significant improvements in computational speed. We make extensive use of the spatial operator methods which have been developed recently for the analysis and simulation of the dynamics of multibody systems. The spatial operators are used to derive the equations of motion and obtain an operator expression for the system mass matrix. An alternative square factorization of the mass matrix leads to a closed form expression for its inverse. From this follows the recursive algorithm for computing the generalized accelerations. The computational cost of this algorithm grows only linearly with the number of degrees of freedom. This is in contrast to conventional constrained dynamics algorithms whose cost is a cubic function of the number of degrees of freedom. For the case of a polypeptide molecule with 400 residues, the $O(\mathcal{N})$ algorithm provides computational speedup by a factor of 450 over the conventional $O(\mathcal{N}^3)$ algorithm. We also describe a simplified method for computing and handling the potential function gradients within the dynamics computations.

0021-9991/93 $5.00