A lattice solid model to simulate the micro-physics of earthquakes: toward 3D large scale simulations

David Place and Peter Mora

QUAKES, Department of Earth Sciences, The University of Queensland, Brisbane, Australia (e-mail: mora@earthsciences.uq.edu.au; place@earthsciences.uq.edu.au, phone: +61-7 3365 2128; +61-7 3365 2176, fax: +61-7 3365 3474).

Abstract

The particle-based lattice solid model developed to study the physics of rocks and the non-linear dynamics of earthquakes is based on molecular dynamics principle. In order to allow large scale simulations to be performed, a high performance must be obtained when implementing the model. This is achieved by refining the model according to the computer architecture and by developing special algorithm to fully use the full potential of the given parallel super-computer. Particles in the model represents grains of rocks where interactions are computed via a neighbors table updated only when new contacts occur. An adaptative time step increment is implemented to minimize the total computational time for a simulation. To compute friction between particles a non-linear system is resolved at each time step using a modified Newton algorithm. Except for the computation of the frictional forces, parallelism is obtained by distributing the work load of each loop of the program on the processors. However this involves time consuming communications and on average the program is parallel at 80%. To obtain a better speedup, the particles data must be explicitly distributed on the processors where calculations on the different sub-sets of the model are done separately with a minimum of communications (as it is the case for the computation of frictional forces). It is expected that a parallelism of 90% at least in average would be achieved. The continuous refinement of the model and its adaptation for parallel computer will allow large scale 3D simulations to be performed.

Introduction

The introduction of massively parallel computers marked the beginning of an era when large scale numerical simulations of numerous physical processes became feasible. In 1993 molecular dynamic simulations involving up to 100 million particles become feasible (Lomdahl, 1993[2][3]). The lattice solid model (Mora and Place, 1994[7]; Mora and Place, 1998[6]) is based on molecular dynamics principles and is capable of simulating the physics of rocks and the non-linear dynamics of earthquakes including processes such as fracturing, frictional heating and fault gouge development. The model involves particles with short range interactions representing grains of rocks.

The earlier version of the model involving simple particle interactions (interacting through elastic-brittle bonds) was able to simulate up to 4 million particles (using 2 G-bytes of memory). Because the modelling aims to simulate geological processes, the time scale is much larger than in molecular dynamics. Whereas in molecular dynamics a few thousands of time steps may be sufficient, the lattice solid model simulating 4 million particles will typically require few million time steps. Even so, the rates of movements of model blocks of earth material is many order of magnitude faster than real rates.

Instead of memory, it is mainly the computational time which limits the size of the model. Furthermore, the incorporation of more realistic interactions such as intrinsic friction between particles requires more computation, further reducing the maximum model size that can be simulated. Efficient large scale simulations are needed to enable fault systems to be modelled or realistic surface roughness of faults to be specified, and ultimately to allow three dimensional simulations to be performed.

In order to achieved high performances on a parallel computer, different aspects of the implementation must be carefully addressed: The first aspect is the most important and concerns the parallelization, where the algorithm must be modified to allow the work to be split between the different processors. The second aspect is related to how the computations are split over the processors: each processor should have an equal amount of work to perform. This can be achieved by dynamically balancing the amount work on each processor (i.e. load balancing). The Third aspect concerns communications between processors. A processor may have to wait for some data to become available (currently used and modified by an other processor) in order to continue its computations. Communication costs can be reduced by mapping or duplicating data such that each processor is only accessing data which is not currently used by other processors, or located on another node. The fourth aspect concerns the programming of the algorithm. Unlike the first three aspects, this is directly related to the architecture of a processor and the global architecture of the parallel machine.

The lattice solid model

The lattice solid model has been described in previous papers (Mora and Place, 1994[5]; Place and Mora, 1999[7]) and is summarized as follows. Particles representing idealized round grains of rocks, are arranged in a regular two dimensional triangular lattice. Each particle is bonded with its six nearest neighbors. Bonds break if the separation between two particles exceeds a given value r_{break}. Bonded particles are attracted to one another if the separation is greater than the equilibrium separation r_0. Bonded and unbonded particles repel one another if the separation is less than r_0.

A simple intrinsic friction is applied when two free particles come in contact (i.e. the particles are close enough to repel one another). This results in a tangential acceleration which opposes slip between the two unbonded particles. The computation of the frictional forces requires the resolution of a non-linear system containing all interacting particle-pairs (Place and Mora, 1999[7]).

Optimisations

To achieved best performances the program must be adapted to the computer architecture. For instance data are distributed along the processor in a such way that communications are minimum. Also the programming is adapted such that the best performance is achieved: for a specific algorithm, loops and instructions of a subroutine is organised in a such way that the maximum number of floating point operation per second is achieved. Although these optimisations allow to decrease significantly the total time necessary for a simulation, they are nevertheless dependant of the computer architectures. Other optimisations are algorithmic optimisation that are independent of the computer architecture and in the present case can decrease by a factor 5 the total time necessary for a simuation. These modifications are summarised in the following.

To reduce the total number of time step required for a simulation, a variable time step increment is used. A large time step increment is used during period of quiescence and a small time step increment when the system undergo dramatic changes such as a synthetic earthquakes event. The time step increment typically varies between 0.2 and 0.04 unit of time depending of the maximum particle velocities.

To compute interaction, particles that are interacting together must be known. Interactions between particles are specified by a neighborhood table. This table is a list of particle-pairs where each particle-pair represent the interaction between two particles. Interaction occurs only within a range of r_0 (when particles came into contact). Particle-pairs in the neighborhood table are composed of all particles within a range of $(1+\alpha)r_0$ where α is typically 0.3. By doing so, the neighborhood table will be up-to-date until particles have moved at least a distance of $\frac{\alpha}{2}r_0$. In 2D simulations these computations represent only 2% of the total computational time.

The computation of frictional forces between particles requires the resolution of a sparse non-linear system. The non-linear system is solved using a newton algorithm. Different methods have been used to solve in parallel the sparse linear system such as a supernodal approach (James Demmel et al., 1995[1]) or using a banded matrix solver. Using these methods, the resolution is effectively performed in parallel only for large linear system. Since our linear system is "small" the resolution cannot be performed in parallel: the number of non-zero elements per column is always very small (10 non-zero element per column on average) and therefore, the work done on each column cannot be parallelized. In order to solve this linear system in parallel, the linear system is split into several sub-systems. In order to preserve all interactions, the sub-systems are overlapped. Overlapping sections of the systems are updated during the newton algorithm. This method does not affect the rate of convergence of the Newton algorithm, hence the speedup depends of the size of the overlap. Depending of the size of the system, this algorithm is at least parallel at 90%.

Conclusions and perspectives

Parallel super-computers with a performance of a few GFlops, provide the ability to simulate the physics of rocks and earthquakes in model sizes which would have been infeasible few years ago using the lattice solid model. Using a precise and efficient

numerical algorithm, and exploiting the capabilities of parallel machines, the lattice solid model is capable of simulating complex interactions involving a million particles. However, the program must be well written in order to obtain the best performance. It is expected that compiler enhancements and increasing computer performance (towards TeraFlops super-computers) will enable three dimensional large scale simulation. These will provide the means to probe fundamental questions in the physics of rocks and earthquakes predictability using the lattice solid model.

Acknowledgments

This research was funded by the Australian Research Council. Supplementary funding was provided by The University of Queensland (UQ), the sponsors of QUAKES. Principle computations were made using QUAKES 12 processor Silicon Graphics Origin 2000. Supplementary and seed computations were respectively made using UQ's Power Challenge, QUAKES Power Challenge and the CM-5 of the French National Centre for Parallel Computing in the Earth Sciences (CNCPST).

References

[1] James W. Demmem, Stanley C. Eisenstat, John R. Gilbert, Xiaoye S. Li, and Joseph W.H. Liu, 1995, *A supernodal approach to sparse partial pivoting.* Technical Report UCB//CSD-95-883, Computer Science Division, U.C. Berkeley.

[2] Lomdahl, P. S., Tomayo, P., Gronbech-Jensen, N., and Beazley, D. M., 1993a, *G-flops molecular dynamics on the Connection Machine 5. in:* Proc. Super-Computing 93, Portland Oregon, Nov. 15-19 (IEEE Computer Soc. Press 1993a)

[3] Lomdahl, P. S., Beazley, D. M.,Tomayo, P. and Gronbech-Jensen, N., 1993b,*Multi-million Particle Molecular Dynamics on the CM5,* Int. J. Mod. Phys. **C 4**, 1075-1084

[4] Mora, P. and Place, D., 1993, *A lattice solid Model for the Nonlinear Dynamics of Earthquakes,* Int. J. Mod. Phys. **C 4**, 1059-1074.

[5] Mora, P. and Place, D., 1994, *Simulation of the Frictional Stick-slip Instability,* Pure Appl. Geophys., **143**, 61-87.

[6] Mora, P. and Place, D., 1998, *Numerical simulation of earthquake faults with gouge: toward a comprehensive explanation for the heat flow paradox,* J. Geophys. Res., **103/B9**, 21,067-21,089.

[7] Place, D. and Mora, P., 1999, *The lattice solid model to simulate the physics of Rocks and earthquakes: incorporation of friction,* J. Comp. Phys., **150**, 332-372