Molecular Dynamics Simulations with IMD

Johannes Roth, Jörg Stadler, Marco Brunelli, Franz Gähler, Jutta Hahn, Martin Hohl, Christof Horn, Ralf Mikulla, Gunther Schaaf, and Hans-Rainer Trebin

Institut für Theoretische und Angewandte Physik, Universität Stuttgart, Pfaffenwaldring 57, 70550 Stuttgart, Germany

Abstract. We describe IMD (**I**TAP **M**olecular **D**ynamics), a software package for classical molecular dynamics simulations on massively-parallel computers. IMD is a general purpose program for short-ranged interactions. It contains special features for the investigation of the mechanical properties of solids. A representative application of IMD to shock waves in binary icosahedral quasicrystals is also given. The simulation has been carried out to demonstrate the feasibility of metacomputing (clustering of massively-parallel supercomputers over large distances).

1 Introduction

IMD is a software package designed for classical molecular dynamics (MD) simulations in two or three dimensions. The basic implementation of IMD and a description of release 1.0 has been published in a previous paper by Jörg Stadler et al.[1]. With the first release it was only possible to simulate particles interacting via pair potentials. Several thermodynamical ensembles were allowed, but the volume had to be kept constant.

Meanwhile IMD has been extended and improved. The current release 1.2[2] allows the particles to interact via pair, three-body and embedded atom (EAM[3]) potentials. The particles can be of an arbitrary number of types with different mass. The interaction potentials are read from tables and are therefore not limited to analytical expressions. The simulations can be carried out in microcanonical (NVE), and canonical (NVT, NPT, and NσT[1]) ensembles. The Nosé-Hoover thermostat is used for the canonical simulations. In addition to MD simulations it is possible to optimize the potential energy of a sample with the microconvergence method[5]. Mechanical boundary conditions allow the sample to be pulled apart or sheared at a constant rate or at certain time intervals.

IMD contains routines to compute self-correlation functions and to generate several crystal or quasicrystal structures. In addition there are special setups for fracture simulations to damp out shock waves emitted from a crack, and in the shock wave simulations to initiate a shock wave.

IMDs target platforms are RISC workstations and massively-parallel computers. It does not perform well on vector machines. IMD is written in C and

[1] σ denotes uniaxial stress.

uses MPI library for message passing. It has been adapted to metacomputing via the PACX[6] library. For online visualization the VolRend[7] package has been extended to VolIMD which is permits communication with IMD.

This paper is organized as follows: In section 2 we will describe online visualization and the generation of pictures with IMD. In section 3 we will talk about metacomputing and in the final section present an example, the simulation of shock waves in quasicrystals

2 Online Visualization and the Generation of Pictures

Usually a MD simulation is carried out, data are collected during the run and stored for analysis *after* the simulation has been finished. With the large massively-parallel supercomputers such a scheme may no longer feasible since in many cases the amount of data is too large to be stored. A solution of this problem is to visualize the data online. More advanced schemes even permit to steer the simulation online but this is currently not supported by IMD although the basic interfaces are implemented already.

For online visualization the simulation is started interactively or in batch as usual. At any time during the simulation it is possible to connect a workstation to the supercomputer running the simulation and to get pictures or histograms. This is simply done by starting VolIMD on your workstation. The communication scheme implemented in IMD works in the following way: The central part of a MD simulation is the main loop for integration of the equations of motions. At a given time interval a function, `check_sockets`, is carried out which connects to a specified TCP/IP port at the workstation and looks whether certain flags have been set. If this is the case, the simulation program reacts as required and sends a picture or histogram of the kinetic or potential energy distribution to the workstation.

The output produced by the online visualization routines are raw data written in binary format. This has the advantage of being faster and more flexible than writing formatted output. Tools are available to convert the binary data into ppm[2]-files. In two dimensions ppm-files can be written directly, but the possibility to alter the pictures afterwards is lost.

IMD supports the generation of histograms of the kinetic and potential energy distribution in the sample. It has turned out that the kinetic energy is useful for example to visualize crack tips or shock waves, whereas the potential energy is sensitive to interface atoms or stacking faults. Since quasicrystals are rather complicated structures where the atoms have quite different potential energies it is also possible to store reference potential energies at the beginning of the simulation and to subtract them later.

[2] ppm: portable pixel map.

3 Metacomputing

Metacomputing is the extension of massively-parallel supercomputing to heterogeneous supercomputer clusters. Several applications are conceivable: if an application is too large for a single supercomputer and the computation center possesses several machines or if an application consists of a part more suitable for massively-parallel systems and another part more suitable for vector machines.

IMD is prepared for metacomputing with the PACX[3] library, developed at the HLRS in Stuttgart[6]. If necessary, PACX simply replaces standard MPI routines by new routines. Thus only minimal changes should be required for an existing massively-parallel program since some MPI routines are not yet implemented in PACX.

On each of the participating supercomputer nodes two processors (PE) are set aside for communication, one for sending and one for receiving. Figure 1 shows the communication between two hosts: The communication within one node is unchanged. But if a PE on one node wants to communicate with a PE on the other node, a two-step process is initiated: first the message is broadcasted locally to the communication PE, then it is sent to the communication PE on the other node and transmitted again over the local network to the destination PE. The communication with PACX was originally lim-

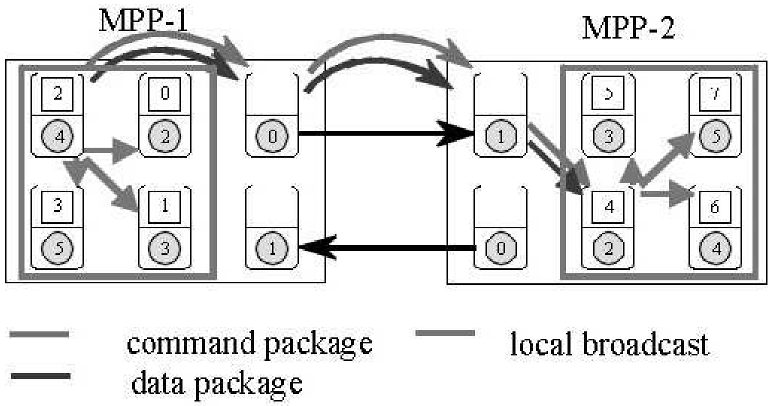

Fig. 1. Communication between two supercomputers.The labels in the squares and circles are the global and local processor addresses, respectively. The straight arrows indicate how the communication processors are connected.

ited to two participants, meanwhile it has been extended to any number of supercomputer nodes. The communication PEs now talk to all other nodes, therefore two communication PE per node are still sufficient.

[3] PACX: **PA**rallel **C**omputer e**X**tension.

3.1 Special Requirements for Metacomputing

IMD has been written for massively-parallel supercomputers where the communication between the PE is very fast and and the communication time is short compared to the time required for a time step. For metacomputing this is no longer the case. The latency time for a message between two partitions on the same T3E is 3.5 ms, between Stuttgart and Manchester it is 45 ms, and between Stuttgart and Pittsburgh it is 75 ms! To avoid wrong results the computations has to wait until the inter-supercomputer communication has been finished. This is very unsatisfactory and a way to interlace calculation and communication has to be found.

The solution for a simulation program with domain decomposition (see Ref. [1]) is rather straightforward: The sample is divided into cells and distributed onto the PEs. Only the content of the surface cells of each PE has to be sent to the neighbor PEs. The loop over all cells on a PE has to be broken up into a central part not necessary to be communicated and into the surface part. During the exchange of the surface part a PE can start with the computations in the central part and then continue successively with the surface part that has been updated already.

4 Shock Waves in Quasicrystals and Crystals

There have been two reasons to study shock waves in quasicrystals: The first is that a large variety of defects can be created in this way. The second reason is that shock waves are suitable for metacomputing simulations since equilibration is unnecessary for such simulations far from equilibrium.

The shock waves were generated in the following way: The velocities of all atoms in a small slab were set to a large value with a direction pointing along the long axis of the rod-like sample.

We have studied two samples: A binary quasicrystal with TI structure[8], consisting of decorated prolate and oblate rhombohedra, and a crystal built of prolate rhombohedra only. Both samples contained about 1 million atoms and are as similar as possible. Their behavior, however, is strikingly different: First of all, the propagation velocity of shock front in the crystal is much larger than the velocity in the quasicrystal (Fig. 2). We assume that the aperiodicity of the quasicrystal decelerates the shock wave.

A second difference is observed for the strength of the material: the crystal is much stronger than the quasicrystal despite the similarity of the structures. Below a shock wave energy $I \approx 3$ in Lennard-Jones units no defects occur in the quasicrystal. Between $I \approx 3$ and $I \approx 13$ shear bands consisting of parallel lines of dislocations show up. Above $I \approx 13$ the sample becomes amorphous. The crystal is still perfect at a shock wave energy of $I = 12$, at $I = 18$ domains of disordered material are created, and at $I > 24$ it becomes amorphous. Shear bands have not been found in the crystal.

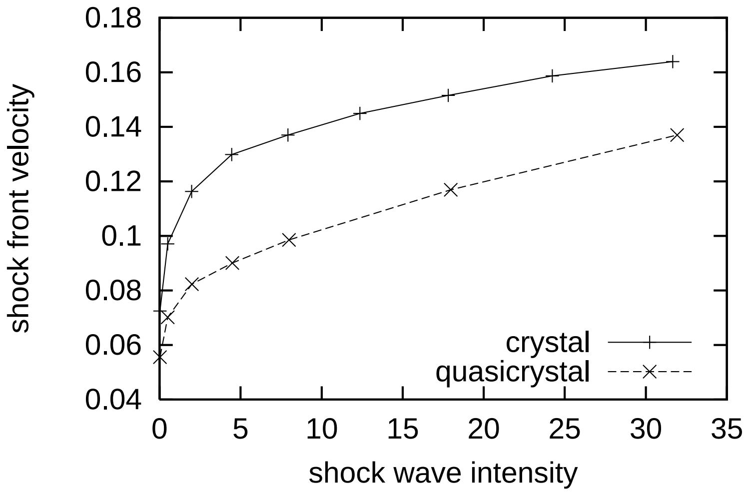

Fig. 2. Velocity of the shock wave (in arbitrary units) vs. strength of the shock wave, expressed in units of the average kinetic energy.

To understand the differences between the strength of the crystalline and the quasicrystalline structure further simulations are necessary.

Acknowledgments

The help of the PACX group at the HLRS Stuttgart, led by Michael Resch and Edgar Gabriel, is greatfully acknowledged.

References

1. Stadler J., Mikulla R., Trebin H.-R. (1997) IMD: A Software Package for Molecular Dynamics Studies on Parallel Computers. Int. J Mod Phys C 8:1131
2. www.itap.physik.uni-stuttgart.de/~johannes/imd-home.html
3. Daw M.S., Baskes M.T. (1984) Embedded Atom Method: Derivation and Application to Impurities, Surfaces and other Defects in Metals. Phys Rev B 29:6443
4. Allen M.P., Tildesley D.J. (1987) Computer Simulation of Liquids, Oxford Science Publications, Oxford
5. Beeler J.R. (1983) Radiation Effects Computer Simulations, North Holland, Amsterdam, p 27
6. www.hlrs.de/structure/organisation/par/projects/pacx-mpi
7. www.uni-stuttgart.de/RUSuser/vis/People/roland/volrend/volrend.html
8. Roth J., Schilling R., Trebin H.-R. (1990) Stability of Monatomic and Diatomic Quasicrystals and the Influence of Noise. Phys Rev B 41:2735