

# IMD: A Typical Massively Parallel Molecular Dynamics Code for Classical Simulations – Structure, Applications, Latest Developments

Johannes Roth

**Abstract** We give a short description of IMD, a classical molecular dynamics package for the simulation of condensed matter. The properties of molecular dynamics simulations will be given with examples of their implementation in IMD. We further report on multi-scale simulations with IMD, the determination of accurate interactions with `potfit` and the porting of IMD to GPUs.

## 1 Introduction

Molecular dynamics (MD) simulations on the atomistic level play an important role today in science and industry. It can be applied – in different ways certainly – from the statistical mechanics of sub-atomic particles up to the most accurate calculation of the orbits of planets or the rings of Saturn. Here, however, we will concentrate on molecular dynamics simulations in condensed matter physics. We start with an explanation of the components of a molecular dynamics simulations and used that for the introduction of the implementation of MD in the **ITAP Molecular Dynamics** package IMD. For more details see the basic publications [16, 22] and the webpage [imd.itap.physik.uni-stuttgart.de/userguide/imd.html](http://imd.itap.physik.uni-stuttgart.de/userguide/imd.html). We will then elaborate a little about one of the most important parts, namely of the modeling of the interactions. Since we want to apply MD to atoms where the interaction is mediated by electrons through metallic, covalent, of ionic bonding, one should solve the problem by quantum mechanics. The drawback is that this would limit the size of the simulations from several hundred up to a few thousand atoms, or to some hundred thousand particles if we use tight-binding methods. Our goal, however, is to simulate big systems beyond millions of particles for up to  $\mu\text{s}$  to be able do deal

---

J. Roth (✉)

Institut für Funktionelle Materialien und Quantentechnologien, Universität Stuttgart,  
Stuttgart, Germany

e-mail: [johannes@itap.physik.uni-stuttgart.de](mailto:johannes@itap.physik.uni-stuttgart.de)

with long-range effects. We do not want to discuss the applicability of classical MD simulations in detail but refer the reader to the Bohr-Oppenheimer approximation. If applicable, we still have to model the interactions of our sample which can be done with *potfit*, a program that fits the forces, energies and stresses of the classical interactions to quantum mechanical simulations of a database of small samples.

The paper is organized as follows: we start with classical molecular dynamics simulations, then we introduce *potfit*. We will further give a number of recent examples of the application of IMD. Then we will give a short account on parallelization and benchmarking IMD. We will continue with the attempts to port IMD to GPUs. We will end with a comment on very big simulations.

## 2 Classical Molecular Dynamics Simulations

The basic idea of classical molecular dynamics simulations is simple: if there are  $N$  particles, integrate the  $3N$  Newtonian second order equations of motions or equivalently the  $6N$  Hamiltonian first order system of equations of motion. Since this is an initial value problem of ordinary differential equations, we have to specify in the case of point-like particles without internal structure  $6N$  starting values which are  $3N$  coordinates and  $3N$  velocities or momenta.

### 2.1 The Molecular Dynamics Steps

In the next sections we will describe the components of a MD simulation. Details of the different steps can be found in the basic book by Allen and Tildesley [2]. Some parts like storing data on tapes are quite outdated, but the major ideas presented in the book are still valid. The examples and features described are those available in IMD [16,22].

#### 1. First step: Initial conditions

The coordinates are typically given by the atom positions in the sample under study. If it is a fluid or gas, the coordinates might be generated by a random number generator. For a crystal, the coordinates are found in crystallographic databases or may be determined from real experiments. Often the coordinates are obtained from model systems for which structural properties or phase diagrams are to be studied.

Normally, the velocities or momenta of the particles are not known. Thus they are typically generated with a random number generator. Since random number generators produce homogeneous distributions of random numbers, they have to be converted to yield the desired Maxwell-Boltzmann distribution (see for example [2]). At the same time they are scaled to give the desired temperature via the equipartition theorem

$$2E_{\text{kin}} = 3Nk_{\text{B}}T$$

where  $E_{\text{kin}}$  is the kinetic energy,  $N$  the number of particles,  $k_{\text{B}}$  is the Boltzmann factor and  $T$  is the instantaneous temperature.

## 2. Second step: Interactions

For a long time, when computers were expensive and small, these were model interactions like the Lennard-Jones pair potential. Such interactions are still applied in statistical mechanics where people are interested in generic features of interactions. If properties of materials should be studied with high accuracy, the method of choice is for example force-matching. Ab-initio calculations are carried out to obtain forces which are then fitted to obtain a classical interaction for the MD simulation. This can be done with *potfit* for example and will be explained in more detail in Sect. 3. In organic chemistry and soft matter physics especially standardized two-, three-, and four-body interactions called force-fields exist for modeling bond lengths, angles, and dihedral angles.

## 3. Third step: Boundary conditions

The surface to volume ratio is important even for the largest simulations in three dimensions. Thus one has to specify boundary conditions. If surface effects are negligible, open or free boundaries are applied, which effectively means that a cluster of particles is simulated. If surface effects should be avoided, periodic boundary conditions are applied, which means that the left and right, upper and lower and top and bottom surfaces are pairwise identified and if a particle leaves the box on one side it will enter on the opposite side again. Thus there is no boundary at all and the simulation box is a torus. In solid state physics and materials sciences it is often of interest to deform the sample by straining it for example. This can be done by fixed boundary conditions, where the atoms at the surface are moved according to a given trajectory or an applied force. Finally, there are more advanced boundary conditions, for example the Lees-Edwards boundary conditions [2] which allow to imprint a flow on the sample without applying forces.

## 4. Fourth step: Integrators and linked lists

While there are basically two formulations of the standard equations of motion for classical particles in a micro-canonical ensemble with constant NVE ( $N$  number of particles,  $V$  volume,  $E$  total energy), namely the Newtonian and Hamiltonian, there is an arbitrary number of integrators. The Hamiltonian equations of motion are

$$\frac{d\mathbf{p}_i}{dt} = -\nabla V(\mathbf{r}_1, \dots, \mathbf{r}_N)$$
$$\frac{d\mathbf{r}_i}{dt} = \frac{\mathbf{p}_i}{m_i}$$

with the interaction potential  $V(\mathbf{r}_1, \dots, \mathbf{r}_N)$  depending on all coordinates  $\mathbf{r}_i$  and  $\mathbf{p}_i$  the momenta of the  $N$  particles. Typically this  $6N$  dimensional system of differential equations is solved by discretizing the time into steps small enough that the basic vibrational frequencies and the fastest motions can still be represented.

Today typical choices of the integrators are the modifications of the so-called Verlet- and leap-frog algorithms. Principally they are inferior to the older Gear-predictor-corrector algorithms or to Runge-Kutta-type integrators, but they have many advantages: they require less storage, which is especially useful for communication on massively parallel supercomputers. Furthermore, they are similar to so-called symplectic integrators which respect the inherent symmetry of the equations of motion and are thus more stable even if their nominal accuracy is lower than that of higher order integrators. A typical leap-frog integration scheme looks like this:

$$\begin{aligned}\dot{\mathbf{p}}_i(t + \Delta t/2) &= \dot{\mathbf{p}}_i(t - \Delta t/2) + \mathbf{f}_i(t) \cdot \Delta t \\ \dot{\mathbf{r}}_i(t + \Delta t) &= \dot{\mathbf{r}}_i(t) + \frac{1}{m_i} \mathbf{p}_i(t + \Delta t/2) \cdot \Delta t\end{aligned}$$

It is called leap frog since the evaluation of positions and coordinates are shifted by half a time-step. First the force  $\mathbf{f}_i(t)$  for each particle  $i$  is computed and added to the previous momenta  $\dot{\mathbf{p}}_i(t - \Delta t/2)$ , then the positions  $\dot{\mathbf{r}}_i(t)$  are updated and the new forces  $\mathbf{f}_i(t + \Delta t)$  will be calculated.

The most time consuming part of the simulations is the evaluation of the forces. If the potential  $V(\mathbf{r}_1, \dots, \mathbf{r}_N)$  is approximated in the most simple approach by pair potentials  $V(\mathbf{r}_i, \mathbf{r}_j)$  which act between particle  $i$  and  $j$ , then the complexity of the algorithm is still  $O(N^2)$  and uses 80% of computation time or more. If the interactions can be made short-ranged by cutting them off at certain distance, then it is possible to introduce linked lists for the administration of the interactions (see [22] for details on IMD.) Typical examples are metals. Long-range interactions between ions or polar molecules like water have to be treated in a different way (see Sect. 3.1). For the linked lists the simulation box is divided into cells with the size of the interaction range. Each atom is assigned to a cell which can be done by local information only. Each atom can only interact with its neighbors in the same cell or in neighboring cells due to the cutoff. Together with actio equal to reactio this leads to 13 partner cells in three dimensions. Now the complexity of the algorithm is only  $O(N)$ , and the goal is to reduce the pre-factor, i.e. the number of particles in each cell. Especially in the case of three-body interactions  $V(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k)$  and many-body interactions this is advisable: the cells are not used to compute directly to compute the forces but to determine the interaction partners. If a certain “skin” is added for particles that are close to interaction, then this list can be recycled and need not be updated at every time step. Such tables are called Verlet-tables [2] although they were originally introduced in a different way which had still an order of  $O(N^2)$  complexity.

### 5. Fifth step: Influence from the outside

Up to now we have studied an isolated micro-canonical  $NVE$  ensemble. If constant temperature as in a canonical  $NVT$  ensemble or even constant pressure  $NPT$  should be simulated, then the integration scheme has to be modified. Further extensions are required if the sample is stressed, deformed, or if flow is applied. Shock waves [13–15] or laser irradiation [19, 20] are other influences which require special treatment.

The method of choice to simulate constant temperature is the Noseé-Hoover thermostat. It can be shown that it leads to a canonical ensemble in limit of infinitesimal time steps. First the instantaneous temperature  $T(t)$  is determined by applying the equipartition principle

$$\frac{3}{2}Nk_B T(t) = \sum_i^N \frac{\mathbf{p}_i^2}{2m_i}.$$

Then system is derived towards the desired temperature  $T$  by feedback applied to the forces:

$$\dot{\mathbf{p}}_i = -\nabla_i V - \eta_T \mathbf{p}_i$$

where  $\eta$  is determined by a new differential equation

$$\dot{\eta} = \nu_T \left\{ \frac{T(t)}{T} - 1 \right\}.$$

The strength of the coupling is determined by the frequency  $\nu_T$  which can be determined from the Einstein frequency and its relation to the average force (for details see [2]).

The simulation of constant pressure works in a similar way. Here we start from the pressure equation for the instantaneous hydrostatic pressure  $P(t)$  of an interacting material:

$$P(t) = \rho k_B T + W/V$$

with density  $\rho = N/V$ , volume  $V$  and virial

$$W = \frac{1}{3} \sum_i^N \mathbf{r}_i \cdot \mathbf{f}_i.$$

The desired pressure  $P$  is again achieved by feedback, not applied to positions and momenta

$$\begin{aligned} \dot{\mathbf{r}}_i &= \frac{\mathbf{r}_i}{m_i} + \xi_T \mathbf{r}_i \\ \dot{\mathbf{p}}_i &= -\nabla_i V - \xi_T \mathbf{p}_i. \end{aligned}$$

The coupling parameter  $\xi_T$  is much more difficult to determine than  $\eta_i$ . In principle it should not alter the frequency spectrum of the simulation. In praxis a first good choice is to set  $\xi_T$  equal to  $\eta_T$ . Then a few tests with a variation of the parameter by a magnitude up or down will show if the coupling will work correctly. Although the coupling parameters are in principle temperature

dependent, we observe that the same parameters can be used at least in the whole solid range of the phase diagram. The method can easily be extended to constant stress simulations of solids with uniaxial loading for example.

Intrinsic flow can be simulated by the Lees-Edwards boundary conditions. The equations of motions should also be modified as given here in two dimensions:

$$\begin{aligned}\dot{x}_i &= \frac{p_{xi}}{m_i} - \gamma y_i & \dot{y}_i &= \frac{p_{yi}}{m_i} \\ \dot{p}_{xi} &= -\nabla_{xi}V + m_i \gamma \dot{y}_i & \dot{p}_{yi} &= -\nabla_{yi}V\end{aligned}$$

The strength of the flow is determined by  $\gamma$ .

## 6. Sixth step: Data evaluation

MD simulations will directly yield potential and kinetic and thus the total internal energy by summing over all particles. To compute the free energy is not so trivial since it is not a mechanical quantity depending on coordinates and momenta. Methods to compute the free energy are given for example in [2]. Further data which are directly available are total and local stresses and displacement fields. Elastic constants can be computed by systematically deforming the sample along the required modes. Transport coefficients can be obtained from equilibrium simulations via the Green-Kubo relations or by non-equilibrium simulations, for example by applying a temperature gradient. Correlation functions are obtained if the required data are added up during simulations. These and many other observables are implemented in IMD together with utilities to evaluate them. Diffraction patterns for example can be calculated from the correlation functions through fast Fourier transform.

## 7. Seventh step: Visualization

Many three-dimensional processes that occur in the bulk and dynamical procedures can only be detected by modern visualization methods. There are many programs today which allow the direct rendering of atoms like VMD ([www.ks.uiuc.edu/Research/vmd](http://www.ks.uiuc.edu/Research/vmd)), ovito ([www.ovito.org](http://www.ovito.org)) or MegaMol ([svn.vis.uni-stuttgart.de/trac/megamol](http://svn.vis.uni-stuttgart.de/trac/megamol)) for example. Typically these programs allow to color code the particles according to selected observables. IMD allows to preselect the particles to be visualized, for example only those which are at the boundary or belong to a defect or form a cluster close to a crack surface.

All the visualization programs also allow to produce movies from sequences of the particle configurations. The movies are especially helpful for the determination of dynamical processes.

Molecular dynamics simulations can be used to study equilibrium problems, like structure as a function of pressure and temperature or the stability of grain boundaries. But they can also be used to determine transport coefficients from fluctuations like diffusion constants or flip frequencies for example. Beyond that we run non-equilibrium simulations close to equilibrium, where the relaxation time is very short. This includes for example the study of plastic deformations, dislocation

motion, crack propagation, shock waves or laser ablation, to name only a few processes where IMD has been used.

### 3 Realistic Interactions and *potfit*

If real properties of materials shall be reproduced with great accuracy it is unavoidable to use the best available interactions. Typically these are quantum mechanical ab-initio descriptions. But for mechanical studies for example samples with several million or hundred million atoms are required which are way beyond the size that can be simulated by ab-initio methods. Therefore a program was developed in close relation to IMD named `potfit` ([potfit.sourceforge.net/wiki/doku.php](http://potfit.sourceforge.net/wiki/doku.php)) [5]. The idea is the following: compute forces, energies, stresses and so on by ab-initio calculations and fit classical interactions applicable to molecular dynamics simulations to match them.

The first step is the creation of a database of samples tractable with ab-initio calculations. This is a crucial step, since the members of the database have to represent the space of application intended. For example if cracks have to be simulated, then the database should contain samples with open surfaces, if high temperatures have to be simulated, then the database should contain samples from classical high temperature simulations, and so on.

The second step is the selection of the interaction. Currently available are simple pair interactions, embedded atom potentials and angular dependent potentials for metals and electrostatic interactions. These have the same format as in IMD. Further interactions can easily be added.

The third step is the choice of the interaction representation. You can choose between splines, where you have to specify the number and position of the supporting points together with the weight, or you can choose a functional form where then the parameters of the representation are fitted.

The last step is the choice of the fitting method: conjugate gradient methods and evolutionary algorithms are possible.

The data that can be fitted are forces, energies and stresses, and each variable can be weighted as desired.

The main problem with the free spline is that it requires data for a good fit where none are available: especially in crystals atoms are placed at discrete atom shells with large gaps in between. Even for heavy distortions these gaps cannot be sampled. Thus the fitted interaction may take a weird shape. But if it is used in simulations, then the particles will find these intermediate ranges which lead to strange results.

While force-matching works well for monatomic and binary samples it becomes less useful for ternary compounds where the fitting space gets too large or the functional shape of the interactions might not be representative enough of the true interactions.

### 3.1 Long-Range Interactions

As noted in Sect. 2.1 it is not possible to cutoff long-range Coulombic or dipolar interactions directly. The most accurate method is the Ewald summation. It cannot be used for large simulations since its complexity is  $O(N^{3/2})$ . Hierarchical methods, FFT and multipole methods reach  $O(N \log(n))$  or close to  $O(N)$ , but they typically do not fit very well into our molecular dynamics scheme with domain decomposition for parallelization (for details see [16, 22]). Wolf et al. [26] have developed a method which is applicable to solids and liquids, where the charges are compensated and thus the interactions fall off fast enough. This method has been implemented in IMD [7] and applied successfully. If the cutoff complies to certain conditions as described in [7], then the Wolf summation is an  $O(N)$  method with a cutoff more than twice the typical cutoff for short-range interactions, and thus rather costly. But it is still up to three orders of magnitude faster than the pure Ewald summation.

Subsequently we have extended the method to the simulation of induced dipoles as it is required for oxides. The model of Tangney and Scandolo [23] has been used for the description of the interaction. Again the implementation has been tested and applied for several studies of silica, alumina, and magnesia [3, 4, 7, 10]. The original parameters which were used by Tangney and Scandolo together with the Ewald summation lead to good results also with the Wolf summation, but there was still an improvement after re-fitting the parameters directly for the Wolf summation.

For the Tangney and Scandolo model the static polarization and the charges of the ions are given from literature or from force-matching and are thus constant during simulation. This is no longer possible if interfaces between metals and metal oxides shall be simulated since the atoms in the metal are neutral while ionized in the oxide, and the charge will vary across the interface. Streitz and Mintmire proposed a model [21] that can deal with this situation. The charges become variable and are determined by minimization of the chemical potential. While Streitz and Mintmire compute the chemical potential by inverting a huge matrix which is rather time consuming and not applicable to large systems, we have implemented a conjugate gradient algorithm which is much better since we have to find a parabolic minimum.

The problem with the chemical potential is that it requires a minimization of the whole system which is very costly on parallel computers. Thus we will have to use external libraries to tackle this problem. Since the chemical potential will vary only close to the interface it might be possible to find a more local treatment as for example in [9] for impurities.

## 4 Some Examples of Recent Simulations

IMD has been applied to many classical problems of materials sciences and solid state physics. We will not recall them here. We will rather present some applications where IMD has been used in a multi-scale environment. One of the older examples



was a simulation by Buehler et al. [8], where the very complicated and time-consuming ReaxFF interactions have been calculated on the fly in an molecular dynamics simulation with IMD. Another application was the determination of the dynamical structure factor for  $Mg_2Zn_{11}$  intermetallics [11]. A long simulation of the material was run with IMD to produce a trajectory as input of the nMoldyn program ([dirac.cnrs-orleans.fr/plone/software/nmoldyn](http://dirac.cnrs-orleans.fr/plone/software/nmoldyn)) which could then be applied to obtain the phonon dispersion relations.

Another case was even more complicated.  $CaCd_6$  is a material built of clusters including tetrahedra which show a phase transition from oriented to rotating. First an accurate interaction was determined by ab-initio calculations and force matching. Then classical simulations were run with IMD to obtain the energy differences between different correlated orientations of neighboring tetrahedra. Since MD was still too slow to study the phase transition directly, the data were used to set up a Monte-Carlo simulation with the transition energies obtained from the MD simulations. With Monte-Carlo the behavior of the system could be determined as a function of temperature and the phase transition could be observed at a temperature which is compatible with experimental observations [6].

As a last example we want to mention laser ablation simulations. Here the problem is that the interaction with the laser and the heat conduction is governed by electronic processes which cannot be described by classical molecular dynamics. But the fast heat conduction requires huge samples on the other hand. There is a two-temperature continuum model available with separate temperature for the electrons and the lattice or atom cores. This model works rather well if solved by finite difference methods and can predict many properties of laser ablation. However, it cannot give information on the atomistic level like the creation of defects, the production of gas and drops, and so on. We have combined the two-temperature model with MD, thus solving the electronic part with finite differences and the atomistic part by molecular dynamics simulations [19, 20]. We have been able to predict melting depths and ablation thresholds. For the simulation of the ablation plume composition, however, the electronic heat conductivity is still too fast and has to be switched off. Nonetheless, the results compare rather well to experiments.

## 5 Parallelization

IMD is parallelized with MPI and domain decomposition as described in [16, 22]. While this scheme works very well for samples with periodic boundary conditions it has some drawbacks for samples with open surfaces like crack propagation, indentation, shock waves or laser ablation. A more dynamical load balancing scheme has not been attempted to be implemented since the load distribution depends strongly on the kind of simulations. Meanwhile an OpenMP parallelization level has been added especially for cases where MPI does not work as for Ewald summation or certain correlation functions in conjunction with Fourier transforms. It turns out that OpenMP leads to increasing performance only up to four to eight

compute cores, depending on the machine. A serious problem is that the compute cores may change their memory location from thread to thread, which could be avoided by setting environment variables.

In conclusion we can state that the MPI version yields always the highest performance, irrespective of shared or distributed memory. We found no case where OpenMP led to a big improvement of performance.

## 6 Benchmarking IMD

IMD was written especially for massively parallel computers. Benchmarks have been published in [16,22]. Although the numbers have changed, the overall behavior is still valid [17,18]. This is a nearly perfect weak scaling and a good strong scaling. Recently this has been confirmed by Cray who tested IMD on the then Top500 machine Jaguar with up to  $1.3 \times 10^{11}$  particles and on the HLRS Hermite. If about a million particles are available per compute unit, then the degradation of performance is about 10% up to the full machine. IMD could reach a considerable amount of the peak performance of the machines without special adaptation or libraries.

## 7 Porting IMD to GPUs

In recent years it has been realized that graphics processing units (GPUs) yield a huge computing capacity after the advent of CUDA, a language which allows to program GPUs efficiently, the topic has gained speed. Nvidia, one of the producers of graphics cards and the creator of CUDA promises speedups up to 100-fold. In reality the speedups are much lower, often they are much below 10. There are MD programs that are written especially for GPUs, like HALMD ([halmd.org](http://halmd.org)) or HOOMD ([codeblue.umich.edu/hoomd-blue](http://codeblue.umich.edu/hoomd-blue)) which are very fast and take heavy advantage of GPUs, but they are rather specialized for certain purposes. On the other hand, there are general purpose programs which are very powerful like lammps (<http://lammps.sandia.gov/>), GROMACS ([www.gromacs.org/](http://www.gromacs.org/)) or IMD which are difficult to port to GPUs since the requirements for different modes of operation can be very different. Moreover, it is nearly impossible to adapt a given data structure such that it possesses similar good performance on CPUs and GPUs.

### 1. Analysis of IMD properties

IMD does not apply external libraries. This is an advantage if compared for example to `pasimodo` which relies heavily on external libraries ([www.itm.uni-stuttgart.de/research/pasimodo/pasimodo.de.php](http://www.itm.uni-stuttgart.de/research/pasimodo/pasimodo.de.php)). It turned out that it is nearly impossible to port this program to GPUs since the workload is spread quite homogeneously over all subroutines.

IMD has been analyzed and ported partially by a group in Poland under the leadership of Rudniki [24]. IMD uses tabulated potentials and none of the special

functions which are available on GPUs. Thus it cannot take advantage of the special GPU features. IMD in general has a rather low arithmetic intensity of computations. There are only 173 instructions per byte fetched. Thus the performance limits are the low utilizability of the graphics card, redundant computations and the small number of registers available. There is, however, no memory bandwidth problem as it is often the case.

The main computation work load in IMD is the force computation which needs 80–98 % of the run time. Thus it is only one single kernel which has to be ported. After optimization, the force computation kernel requires only slightly more than 50 % of the run time.

## 2. **Adaption of IMD**

A thread on the GPU is the computation task of a single atom. A thread block is a single cell of interacting atoms. The data to be sent are the coordinates, atom types, embedding energy and derivative in the case of embedded atom potentials. The received data are forces, energies, virial, and the embedding energy and its derivative.

## 3. **The algorithm on the GPU**

First load the cell number, position, atom count and number and the position of neighbors. Then each thread block loads the atom data to shared memory. The threads compute interactions in the same cell. Next prefetch the data of the neighbor cells, preprocess the non-interacting neighbors. Now the threads compute the interaction with the neighbors. At last return the results to global memory.

## 4. **Optimization IMD**

The data structure has to be changed from “first atom, next atom” to “first coordinate”, “second coordinate” and so on to take advantage of the memory structure on the GPU. Further optimizations are given by sorting for spatial locality, applying actio equal reactio, computing the interaction matrix first and summing it up. The atoms are renumbered such that subsequent atoms are close in space. For the computation with neighboring cells the atoms are renumbered dynamically. The workflow is changed such that CPU and GPU can work in parallel.

## 5. **Speedups**

The ported IMD code has been tested on several configurations, notably on 24 nodes with 2 AMD 6134 (8 Core) and 2 Nvidia 480 GTX cards, with single node Intel E 5620 (8 Core) and 2 Tesla C2050, and on single Laki nodes Intel E 5520 (8 Core) and 1 Tesla S1070 card. The raw speedup of about 100x (as predicted by Nvidia!) is reduced to 7x by geometric constraints. The speedup with respect to one CPU is 20 to 40x, with respect to the 8 Cores available it is 2.5 to 4x. Tesla and GTX are similar, and there is no notable difference between consumer and professional cards which means that CRC control does not play a role.

The speedup is comparable to lammps which is a similar general purpose MD code. It is certainly inferior to HALMD or HOOMD for the reasons noted above.

Up to now only pair interactions and the rather similar embedded atom interactions have been ported. This means that simulations of metals are possible. These interactions do not require Verlet tables which are a major problem for the porting to GPUs, since they are realized as pointers in IMD and thus cannot be transferred to GPUs in a simple manner. Porting further parts of IMD is work in progress.

## 8 A Comment on World Records Molecular Dynamics Simulations

Recently the world record for atomistic MD simulations was raised to  $4.125 \times 10^{12}$  particles by Vrabec et al. [25]. The previous record stood at  $1 \times 10^{12}$  [12] for about 7 years. If a rather big time step of 1 fs is assumed for the simulation of such a sample with an edge length of about  $5 \mu\text{m}$ , furthermore a typical velocity of sound of 500 m/s, then the simulation has to last of the order of 1 million time-steps to allow the sound and thus the information of the local state of the sample to cross the simulation box once! The simulations, however, were run for 50 time-steps only. In contrast, the simulations of Abraham et al. [1] in 2002 of a billion atoms were run for 200,000 time steps, thus allowing sound to cross the sample about 3–4 times during simulation.

If we look at the size of the samples we find that the only purpose of these records is to fill up the memory of the supercomputers. Continuing the previous trends we should have reached about  $1 \times 10^{14}$  particles already. So the memory has not grown as expected. Looking at the velocity of sound we would need at least several hundred million of time steps which means that we have to run the simulations on the **full** machine for up to a year (the simulations of Abraham et al. lasted 4 days, thus size  $\times$  time-steps was the same as Vrabec et al.'s 11 years later).

We conclude that MD simulations cannot use the full power of the biggest supercomputers effectively, since the number of time-steps has to increase in accordance with the number of processing units (which leads to a larger main memory). More time-steps can only be achieved by faster processing units with higher clock rates. GPUs would certainly help to some extent, but they would not overturn the basic conclusions. Longer time-steps are not applicable, since the size of the time step is given by the basic vibration frequency of the atoms and thus by physics.

## 9 Summary

We have presented the molecular dynamics simulations in general as they are applied in condensed matter physics. We have introduced `potfit` to obtain accurate interactions from ab-initio calculations. A number of recent applications

of IMD to multi-scale simulations have been described. We have further reported the first attempts to port IMD to GPUs.

Although there will be changes in the maintenance, porting and further development of IMD in future we are confident that it is a valuable MD simulation package and will continue to be in future.

## References

1. Abraham, F.F., Walkup, R., gao, H., Duchaineau, M., Diaz De La Rubia, T., Seager, M., Simulating materials failure by using up to one billion atoms and the world's fastest computer: Brittle fracture. Proc. Nat. Acad. Sci **99** 5777–5782 (2002).  
Abraham, F.F., Walkup, R., gao, H., Duchaineau, M., Diaz De La Rubia, T., Seager, M., Simulating materials failure by using up to one billion atoms and the world's fastest computer: Work hardening. Proc. Nat. Acad. Sci **99** 5783–5787 (2002).
2. Allen, M.P., Tildesley, D.J. Computer simulations of liquids, Oxford University Press 1987.
3. Beck, P., Brommer, P., Roth, J., Trebin, H.-R., *Ab initio* based polarizable force field generation and application to liquid silica and magnesia. J. Chem. Phys. **135** 234512 (2011).
4. Beck, P., Brommer, P., Roth, J., Trebin, H.-R., Influence of polarizability on metal oxide properties studied by molecular dynamics simulations. J. Cond. Matt. **24** 485401 (2012).
5. Brommer, P., Gähler, F., Potfit: effective potentials from ab-initio data. Modelling Simul. Mater. Sci. Eng. **15** 295–304 (2007).
6. Brommer, P., Gähler, F., Mihalcovič, M., Ordering and correlation of cluster orientations in CaCd<sub>6</sub>, Phil. Mag. **87** 2671–2677 (2007).
7. Brommer, P., Beck, P., Chatzopoulos, A., Gähler, F., Roth, J., Trebin, H.-R., Direct Wolf summation of a polarizable force field for silica. J. Chem. Phys. **132** 194109 (2010).
8. Buehler, M.J., Dodson, J., van Duin A.C.T., Meulbroek, P., Goddard, W. A., The Computational Materials Design Facility (CMDf): A powerful framework for multiparadigm multi-scale simulations, Mat. Res. Soc. Proceedings (Combinatorial Methods and Informatics in Materials Science), **894**, LL3.8 (2006).
9. Elsener A. Politano, O., Derlet, P.M., Van Swygenhoven, H., Mod. Sim. Mat. Sci. Eng. **16** 025006 (2008).
10. Hocker, S., Beck, P., Schmauder, S., Roth, J., Trebin, H.-R., Simulation of crack propagation in alumina with *ab initio* based polarizable force field. J. Chem. Phys. **136** 084707 (2012).
11. Euchner, H., Mihalcovič, M., Gähler, F., Johnson, M.R., Schober, H., Rols, S., Suard, E., Bosak, A., Ohhashi, S., Tsai, A.-P., Lidin, S., Pay Gomez, C., Custers, J., Paschen, S., de Boissieu, M., Anomalous vibrational dynamics in the Mg<sub>2</sub>Zn<sub>11</sub> phase, Phys. Rev. B **83** 144202 (2011).
12. Germann, T.C., Kadam, K., Trillion-atom molecular dynamics becomes reality, Int. J. Mod. Phys. C **13** 1315–1319 (2007).
13. Roth, J., Shock waves in complex binary solids: Cubic Laves crystals, quasicrystals, and amorphous solids, Phys. Rev. B **71**, 064102 (2005).
14. Roth, J., Shock waves in materials with Dzugutov-potential interactions, Phys. Rev. B **72**, 014125 (2005).
15. Roth, J.,  $\omega$ -phase and solitary waves induced by shock compression of bcc crystals, Phys. Rev. B **72**, 014126 (2005).
16. Roth, J., Gähler, F., Trebin, H.-R., A molecular dynamics run with 5.180.116.000 particles. Int. J. Mod. Phys. C **11**, 317–322 (2000).
17. Roth, J., Karlin, J., Sartison, M., Krauß, A., Trebin, H.-R., Molecular dynamics simulations of laser ablation in metals: parameter dependence, extended models and double pulses, in High Performance Computing in Science and Engineering '12, eds. W.E.Nagel, D.B. Kröner, M.M. Resch, Springer Heidelberg, 2013, in print.

18. Roth, J., Trichet, C., Trebin, H.-R., Sonntag, S., Laser ablation of metals, in High Performance Computing in Science and Engineering '10, eds. W.E.Nagel, D.B. Kröner, M.M. Resch, Springer Heidelberg, 2011, pp. 159–168.
19. Sonntag, S., Roth, J., Gähler, F., Trebin, H.-R., Femtosecond Laser Ablation of Aluminum, Appl. Surf. Sci. **255** 9742 (2009).
20. Sonntag, S., Trichet Paredes, C., Roth, J., Trebin, H.-R., Molecular Dynamics Simulations of Cluster Distribution from Femtosecond Laser Ablation in Aluminum, Appl. Phys. A **101** 559–565 (2011).
21. Streitz, F.H., Mintmire, J.W., Electrostatic potentials for metal-oxide surfaces and interfaces. Phys. Rev. B **50** 11996–12003 (1994).
22. Stadler, J., Mikulla, R., Trebin, H.-R., IMD: A software package for molecular dynamics studies on parallel computers. Int. J. Mod. Phys. C **8**, 1131–1140 (1997).
23. Tangney, P., Scandolo, S., An *ab initio* parametrized interatomic force field for silica. J. Chem. Phys. **117** 8898–8904 (2002).
24. Tredak, P., Lewinski, B., Ligowski, L., Wejranowski, T., Rudniki, W.R., Large scale molecular dynamics simulations of materials on GPU clusters, Preprint
25. Vrabec, J., private communication
26. Wolf, D., Keblinski, P., Philipot, S.R., Eggebrecht, J., Exact method for the simulation of Coulombic systems by spherically truncated, pairwise  $r^{-1}$  summation. J. Chem. Phys. **110** 8254–8282 (1999).