KTH Engineering Sciences

Optimization of VASP simulations relating to radiation damage in materials

Daniel Karlsson and Amanda Rasmussen

danielk5@kth.se, amandara@kth.se

SA114X Degree Project in Engineering Physics, First Cycle

Supervisor: Pär Olsson

Department of Physics
School of Engineering Sciences
Royal Institute of Technology (KTH)

Stockholm, Sweden

TRITA-FYS-2017

May 20, 2017

Abstract

One of the challenges with nuclear reactors is choosing materials that can endure intense levels of radiation. When irradiating a material, the atoms starts a process of ballistic collisions, but it can also lead to radioactive decay or transmutation of the material. The radiation interacts with the materials and causes a wide range of microscopic damages, which over time accumulate to create macroscopic damaging effects such as brittleness, swelling and deformations. With time, the materials can transmute into alloys, and it is of great importance to study irradiation on crystals of different elements.

It is very common to use molecular dynamics simulations for this sort of research. A well known *ab initio* simulation package based on Density Functional Theory (DFT) is VASP. Executing a large simulation cell in VASP is computationally expensive however, so the main focus of this project has been to create a program that will speed up the simulations, without sacrificing too much accuracy.

LAMMPS is another simulation package, but unlike VASP, it is based on classical mechanics. To optimize the simulations, we first predict the trajectories of the atoms in LAMMPS to see if any will experience strong local compression conditions and if so, use a more accurate potential model for those atoms in VASP.

The program written for this task is called LAVAX for LAMMPS VAsp eXchanger. An analysis of the results of simulating radiation damage in a tungsten crystal showed that LAVAX is more efficient by a factor of up to 6.3 compared to running VASP alone.

Acknowledgments

We wish to express our sincere gratitude to our supervisor Pär Olsson, who has given us great guidance through this project. His motivation and patience with us has been an excellent support, and he has provided us with all the knowledge within the computational and materials science field we needed to complete our Bachelor Thesis.

Contents

Chapter 1

Introduction

Nuclear reactors operate under extreme conditions: high temperature, intense levels of radiation and in some cases chemical corrosion. This puts high requirements on the materials used for cladding tubes, reactor vessel, plasma-facing components etc.

The high-energetic neutrons released in the nuclear reactions interact with the materials and cause a wide range of microscopic damages, which over time accumulate to create macroscopic damaging effects such as brittleness, swelling and deformations. [3] The evolution of these damages are complicated phenomena, and ranges on length scales from Ångstroms to meters and on timescales from femtoseconds to years. Besides direct experimental examination, it is also possible to simulate these damages in a computer, which is often faster, cheaper and gives more insight into the process.

Different time/length scales call for different simulation methods. Ab-inito (latin: *from the beginning*) methods requires no previous experimental data (besides the atomic number) and gives the best agreement with reality in comparison with other simulation methods. Ab-initio means that the methods used are based on fundamental principals of nature, such as solving the Schrödinger equation for some system. An example of a non-ab initio method would be molecular dynamics based on classical mechanics, where the atoms are approximated by a semi-empirical potential. The ab initio method used in this work is the popular Density Functional Theory (DFT) method. [13] DFT is however computationally expensive even for moderately sized systems, and almost exclusively requires supercomputer clusters to run a simulation in a reasonable time period.

The objective of this work is to speed up the DFT simulation by using a detailed quantum mechanical potential model in only those parts of the crystal that require it and a less refined model in the rest of the crystal. This will benefit future works and simulations, since it allows for faster execution and thus larger crystals can be analyzed without loss of accuracy.

In this work we have studied radiation damage on a crystal of tungsten, a proposed material for plasma-facing components in fusion reactors. [4]

1.1 Background

1.1.1 Radiation damage in materials

Radiation damage in materials occurs when energetic neutrons collides with the atoms in a crystal. The kinetic energy of the neutrons is directly transferred to some of the atoms. These atoms are called the Primary Knock-on Atoms (PKA). The PKA starts a cascade of ballistic collisions between the atoms in the crystal. The collisions displace the atoms in the crystal, and these displacements leaves a vacancy behind that may be left empty or filled by another atom. [3] The energy of the PKA will spread to the entire crystal as thermal energy and eventually one can measure the defects that remain.

Defects

There are different defects that can occur when a material is exposed to radiation, and a few of them are worth mentioning here. A crystal consisting of one or different atom types can get both Foreign Interstitial Atoms (FIA), substitutional atoms, Self Interstitial Atoms (SIA) and vacancies. The FIA are atoms that naturally occur in the interstitial positions of the crystal (blue atom in Fig. 1.1), such as carbon or oxygen in a metallic lattice. The substitutional atoms (SA) are the ones who occupy the ordinary lattice position (red atom). Both FIA and SA are defects that comes from the crystals composition because of impurities.

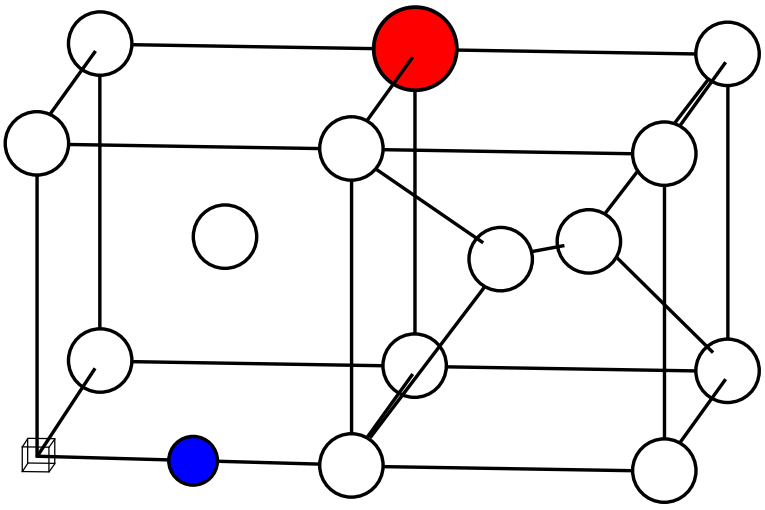


Figure 1.1: Different types of defects in a crystal.

SIA is an extra atom that defects the crystal structure by crowding it's way into the crystal (the dumbbell at position (1.5, 0.5, 0.5)). A vacancy is a missing atom in the lattice (origin). SIAs and vacancies are generated in equal proportion when atoms start to collide into each other. [5]

1.1.2 VASP - Vienna Ab initio Simulation Package

This is one of the two computer programs we have used during this project. VASP is a program for atomic scale materials modeling, and is very common for quantum mechanical (QM) molecular dynamics. VASP utilizes DFT for molecular dynamics simulations and the simulation cell is using periodic boundary conditions. For every ionic step VASP takes, the Kohn-Sham equations are used to reach convergence of the electron density (explained in section 2.1).

Since VASP is using DFT it is computationally expensive, and that is why LAMMPS has an important key role in this project. [13][15]

1.1.3 LAMMPS - Large scale Atomic/Molecular Massively Parallel Simulator

Unlike VASP, LAMMPS is used for classical mechanical molecular dynamics. This is an important difference since classical mechanics is not based on heavy calculations compared to a quantum mechanics. Another difference is that a more accurate model of the potential is used for the quantum mechanical simulation compared to the classical mechanical.

LAMMPS integrates Newton's equation of motion for a collection of atoms that interact via short- or long-range forces with a variety of initial and boundary conditions. LAMMPS uses neighbor lists to keep track of nearby particles for computational efficiency. The atoms are described by a semi-empirical potential model. In this work the relevant potential model is the Embedded Atom Model (EAM), which is suitable for studying metallic crystals. LAMMPS also provide other potential models for studying polymeric, biological, granular and coarse-grained systems. The boundary conditions in LAMMPS can be set to be periodic or fixed. [12][8]

1.1.4 Ovito - Open Visualization Tool

This is a visualization tool for atomistic simulations, where the simulation cell and the entire sequence of events can be seen in a three dimensional space. Multiple particle types can be visualized, and the particles can be color coded for various properties (kinetic energy for instance).

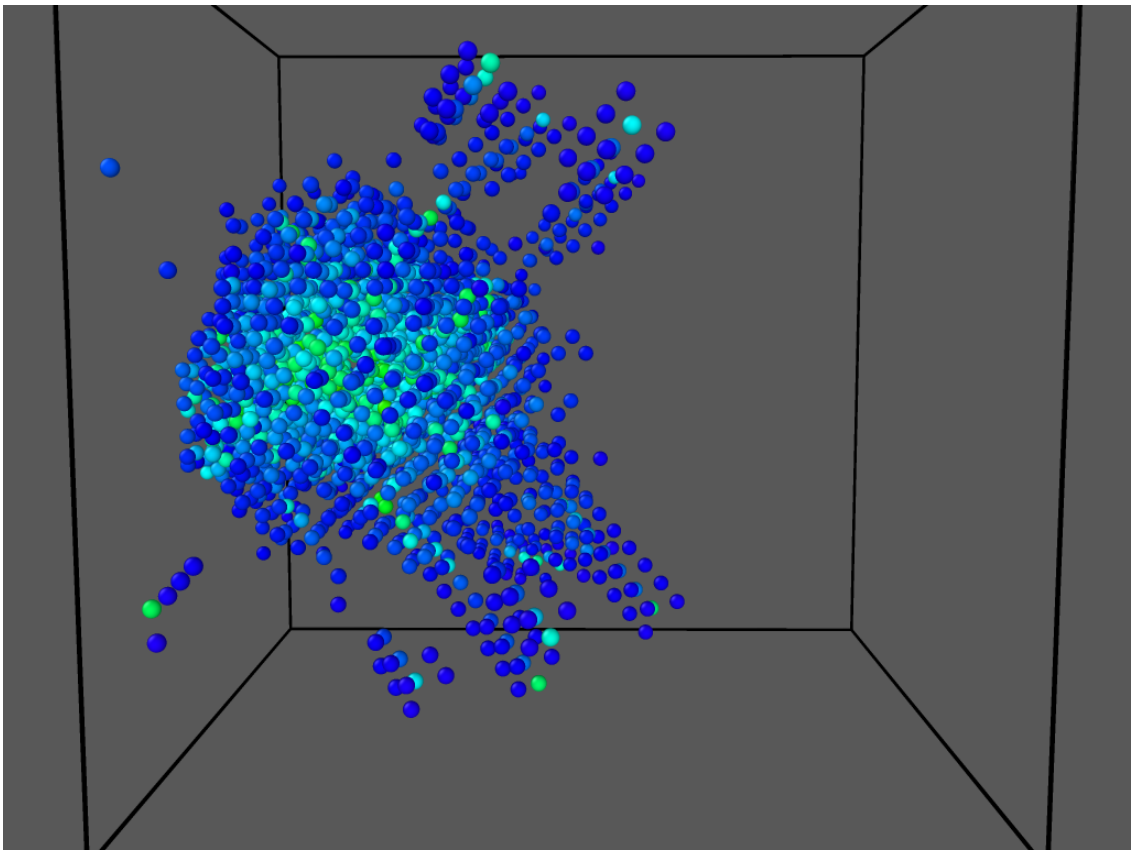


Figure 1.2: Radiation damage visualized in OVITO

1.2 Scope

When studying radiation damage in reactor materials, it is important to use simulations to see what properties a certain material have and how it is affected when it is exposed for radiation. Density Functional Theory publications is becoming more common nowadays compared to 30 years ago for this kind of research. To use DFT calculations for molecular dynamics is starting to become tractable at the more and more powerful supercomputers that are available today. [10] To use a quantum mechanical simulation gives, as mentioned before, the best accuracy on the atomic scale but is computationally challenging, so how can this problem be solved? How can the execution time of a simulation be minimized, and still give a good accuracy of the results?

This study will not attempt to quantify the final state of defects of a crystal, but rather focus only on the performance aspect of simulating radiation damage in VASP.

1.3 Objective

The aim of this study is to write code that exchange data between VASP and LAMMPS. First, LAMMPS will run to predict the trajectories of the atoms in a crystal during some short time interval. LAMMPS will simultaneously build neighbor lists between the atoms within some radius. The neighbor lists will then be used to determine what atoms require a more detailed potential model in VASP. VASP will then be run for the same short time interval. This process will be repeated for a desired number of time steps.

A comparison can be done between the execution of the code and a more accurate, full simulation using VASP. An analysis of the results will tell how much the program differ from running only VASP, and if our code manages to speed up this kind of simulation.

The potential models provided by VASP will also need to be compared to see at what distance they start to deviate from each other.

Chapter 2

Theoretical Background

The code VASP is based on DFT, so it will be explained here briefly so the reader can understand both the theoretical background of DFT and why it is computationally expensive to use for a large number of atoms. LAMMPS has also been used during this project and the theory behind LAMMPS will also be explained.

2.1 DFT - Density Functional Theory

The Density Functional Theory is basically a method to solve the time independent, non-relativistic Schrödinger equation with approximations, because the parameters that are required for solving the Schrödinger equation are computationally expensive. For a system containing M nuclei and N electrons, the Schrödinger Equation can be written as:

$$\hat{H}\Psi_i(\mathbf{x_1}, ..., \mathbf{x_N}, \mathbf{R_1}, ..., \mathbf{R_M}) = E_i\Psi_i(\mathbf{x_1}, ..., \mathbf{x_N}, \mathbf{R_1}, ..., \mathbf{R_M}) \tag{2.1}$$

where $\hat{H}$ is the Hamiltonian of the system.

Hohenberg and Kohn laid the theoretical foundation of DFT in the 1960s by proving that the determination of the ground-state wavefunction (2.1) of the electrons in a system (a function of $3N$ variables in a system containing N electrons) can be replaced by the determination of the ground-state electronic density (a function of only 3 variables). [2]

Born-Oppenheimer approximation

The Hamiltonian can be written as:

$$\hat{H} = T_e + T_n + V_{ext} + V_{ee} + V_{nn} \tag{2.2}$$

where T_e is the kinetic energy of the electrons, T_n is the kinetic energy of the nuclei, V_{ext} is the attractive electrostatic interaction between the nuclei and the electrons, V_{ee} and V_{nn} is the repulsive potential due to the electron-electron and nucleus-nucleus interactions.

The Born-Oppenheimer approximation is made by considering that the nuclei move much slower than the electrons, that we can consider the kinetic energy of the nuclei as zero and their potential energy as constant.[5] With the Born-Oppenheimer approximation the Hamiltonian reduces to:

$$\hat{H} = T_e + V_{ext} + V_{ee} \tag{2.3}$$

which is called the electronic Hamiltonian. Then the solution of the Schrödinger equation reduces to:

$$\hat{H}_{elec}\Psi_{elec} = E_{elec}\Psi_{elec} \tag{2.4}$$

The variational principle

The variational principle is a method to find the ground state energy E_0 by guessing an upper bound energy of the state Ψ.[11] The expected value of the energy E is a functional and can be written as:

$$E[\Psi] = \frac{\langle\Psi|\hat{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle} \geq E_0 \tag{2.5}$$

with

$$\langle\Psi|\hat{H}|\Psi\rangle = \int \Psi^*\hat{H}\Psi dx \ , \quad \langle\Psi|\Psi\rangle = \int \Psi^*\Psi dx$$

To get the ground state energy E_0, the functional must be minimized with the wave functions which depends on the number N of electrons. The functional will also depend on the nuclear potential:

$$E_0 = \min_{\Psi\to N} E[\Psi] = E[N, V_{ext}]$$

Here, $\Psi \to N$ means that Ψ is an allowed N-electron wave function.

The Hohenberg-Kohn theorems

The first Hohenberg-Kohn theorem states that the external potential V_{ext} and the full many particle state are unique functionals of the electron density $\rho(\mathbf{r})$.
The second Hohenberg-Kohn theorem states that the functional delivers the lowest ground state energy of the system when the input density is the true ground state density.[11]

The Kohn-Sham equations

By using the Hohenberg-Kohn theorem the properties of a system can be calculated, but the theorem is due to the ground state density is given. The Kohn-Sham equations is a way of finding the ground state density. [11][5]
Introducing a reference system that is non interacting and has the same density as the real system, we can define the density and the kinetic energy as follows:

$$\rho(\mathbf{r}) = \sum_i^N |\psi_i(\mathbf{r})|^2 \tag{2.6}$$

$$T_s = -\frac{1}{2}\sum_i^N \langle\psi_i| \bigtriangledown^2 |\psi_i\rangle \tag{2.7}$$

But the kinetic energy of the reference system differes from the kinetic energy of the real system. To minimize this difference, one can separate the functional $F[\rho]$ that contains all of the contributions of the kinetic energy:

$$F[\rho] = T_s[\rho] + J[\rho] + E_{XC}[\rho] \tag{2.8}$$

The energy of the real system can be expressed with (2.8):

$$E[\rho] = T_s[\rho] + J[\rho] + E_{XC}[\rho] + E_{Ne}[\rho] \tag{2.9}$$

$$J[\rho] = \frac{1}{2}\int\int \frac{\rho(\mathbf{r_1})\rho(\mathbf{r_2})}{r_{12}} d\mathbf{r_1} d\mathbf{r_2} \quad E_{Ne}[\rho] = \int V_{Ne}\rho(\mathbf{r}) d\mathbf{r} \quad E_{XC}[\rho] = (T[\rho] - T_s[\rho]) + (E_{ee}[\rho] - J[\rho]) \tag{2.10}$$

$J[\rho]$ is called the classical Coloumb interaction, $E_{Ne}[\rho]$ is the functional of the nuclear-electron interaction and $E_{XC}[\rho]$ is the exchange correlation energy. $E_{XC}[\rho]$ is unknown and determined locally by the systems electronic density. Since the Perdew Burke Ernzerhof (PBE) potential is used, superior limits are used for the exchange-correlation energies. [17] We can now use the variational principle and $\langle\psi_i|\psi_j\rangle = \delta_{ij}$ for (2.9). The Kohn-Sham equations can now be formulated:

$$\Big[-\frac{1}{2}\bigtriangledown^2 + V_{eff}(\mathbf{r_1}) \Big]\psi_i = \epsilon_i \psi_i \tag{2.11}$$

$$V_{eff}(\mathbf{r_1}) = \int \frac{\rho(\mathbf{r_2})}{r_{12}} d\mathbf{r_2} + V_{XC}(\mathbf{r_1}) - \sum_A^M \frac{Z_A}{r_{1A}} \tag{2.12}$$

$$\rho(\mathbf{r}) = \sum_i^N |\psi_i(\mathbf{r})|^2 \tag{2.13}$$

To solve the Kohn-Sham equations practically, the first step is to guess the density. Next step is to determine the effective potential (2.12), and then solve the Kohn-Sham equation (2.11) to find a new value of the density. Continue to find a new value of the potential (2.13) and iterate until convergence. The process of solving the Kohn-Sham equations is described in Fig. 2.1:

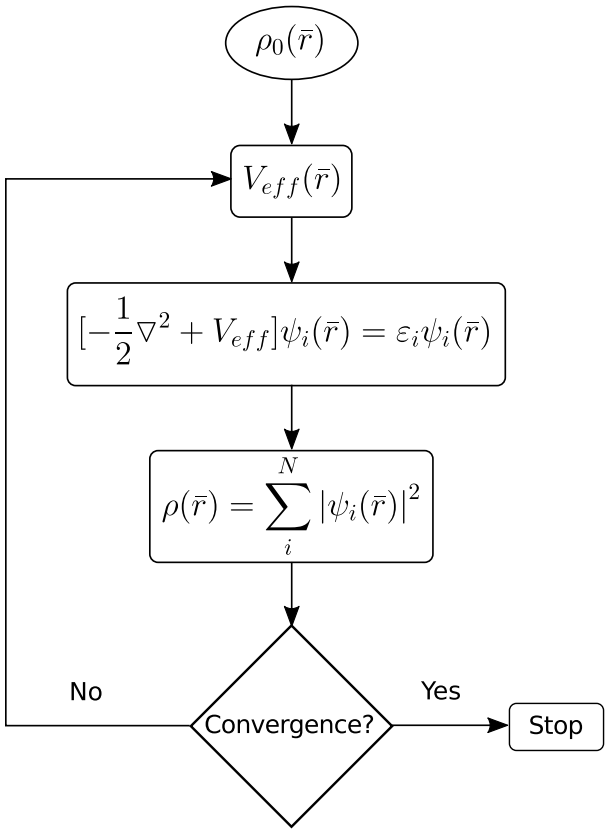


Figure 2.1: Algorithm for finding the ground state charge density in DFT.

VASP use the Kohn-Sham equations for every ionic step as mentioned before, and the computing time of M atoms is proportional to M^3. [16]

2.2 Wave functions

It is necessary to have a good representation of the atoms wave function in DFT. The representation have to be precise in the region of interest, as well as computationally efficient in any part of space. Several methods are available to represent the wave function:

2.2.1 Pseudopotentials

These methods often use the plane waves approximation:

$$\Psi_i(r) = \sum_K c_{i,K} e^{i(k+K)r} \tag{2.14}$$

However, this requires a large basis set to fit the real wave function in the core region, where the wave function varies greatly, in order to be convergent. In areas far away from the core, most of the physical properties are best described by the valence electrons and the core can be considered frozen. To speed up convergence, a smaller basis set is used to fit the wave function in the valence region and is (unrealistically) softened in the core region, as illustrated in Fig. 2.2. The choice of cutoff radius r_c is a trade-off between fast convergence and precision in the core. [5]

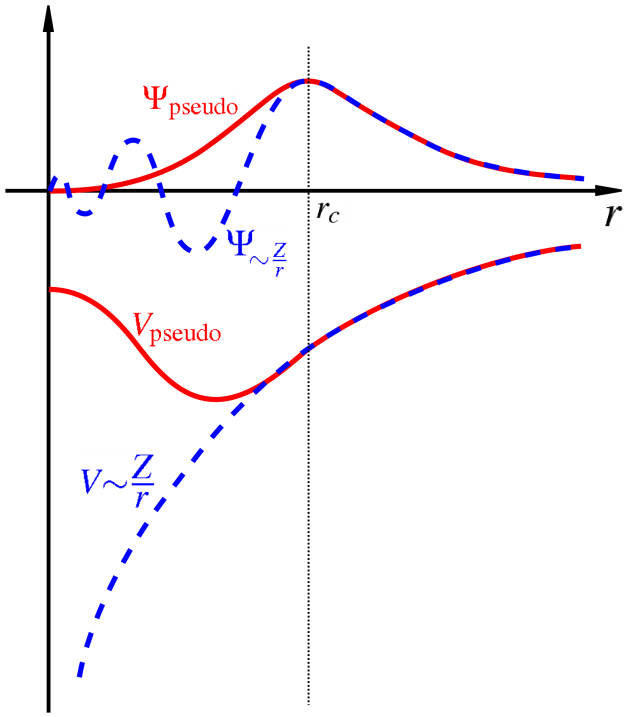


Figure 2.2: When the atoms are far enough apart, it is sufficient to only include the valence electrons in the potential and the core is softened. [6]

2.2.2 Augmented Wave methods

These methods are based on the "muffin-tin" approximation, where the crystal space is divided in to two regions: The interstitial region, where the potential is assumed to be constant and the spherical "muffin" region centered on the atom positions with radius s, where the potential is the average radial value:

$$V(r) = \begin{cases} V(r) & , \ r < s \\ V_0 & , \ r \geq s \end{cases}$$

Since the potential is spherically symmetric, the radial part of the Schrödinger equation becomes, with $u(r) = rR(r)$:

$$-\frac{\hbar^2}{2m}\frac{d^2u}{dr^2} + \left[V(r) + \frac{\hbar^2}{2m}\frac{l(l+1)}{r^2}\right]u = Eu$$

with the full solution for one atom in the region $r < s$:

$$\Psi^{n=1}(r,\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} a_{lm}R_l(r)Y_l^m(\theta,\phi)$$

In the interstitial region where the potential is constant, the wave function is a plane wave series (Eq. 2.14). Here the a_{lm} must be chosen to the wave function is continuous at $r = s$. The wave function for the entire crystal is a linear combination of all the one atom wave functions:

$$\Psi_k \propto \begin{cases} \sum_{l=0}^{\infty}\sum_{m=-l}^{l} A_{lm}R_l(r)Y_l^m & , \ r < s \\ \sum_K c_{k,K}e^{i(k+K)r} & , \ r \geq s \end{cases}$$

[5]

2.2.3 PAW - Projector Augmented Wave method

The Projector Augmented Wave formalism is a complicated pseudoization scheme, similar to the ultrasoft scheme (as described in 2.2.1) but allows for the reconstruction of the real electronic density and the real wave functions, with all their oscillations (as seen in Fig. 2.2). [2] In the valence zone, plane waves are used (as in the ultrasoft scheme), but in the core region, the wavefunction is expressed with the help of atomic orbitals (from the augmented wave method). This method can be considered a hybrid of the two. [5][14]

In this work we have only used PAW potentials, provided by VASP.

2.3 Screened Coulombic repulsion

Ionic repulsive interaction can be described as a screened Coulombic interaction at very short distances. This can be modeled by multiplying the ordinary Coulombic repulsion between the nuclei with a screening function $\chi(r/a)$:

$$V(r) = \frac{Z_1 Z_2 e^2}{4\pi\epsilon_0 r}\chi(r/a)$$

with the condition that $\chi(r/a) \to 1$ when $r \to 0$. Z_1 and Z_2 are the nuclear charges, and a is the screening length. One such parameterization of χ is the Biersack-Ziegler potential, which was constructed by fitting a universal screening function to repulsions calculated for many different atom pairs:

$$\begin{aligned}\chi(x) = 0.1818e^{-3.2x} + 0.5099e^{-0.9423x} \\ + 0.2802e^{-0.4028x} + 0.02817e^{-0.2016x}\end{aligned}$$

where

$$a = \frac{0.8854a_0}{Z_1^{0.23} + Z_2^{0.23}}$$

and $x = r/a$ and $a_0 = 0.529$ Å is the Bohr radius. [1]

It will be a good measure to compare the PAW-potentials provided by VASP with the screened Coulumbic potential at short distances to see if they actually approach each other.

2.4 Classical Mechanics Theory

One of the theories that molecular dynamics simulations can be based on is classical mechanics, Newtons law of motion is used for a given atomic system and is defined as:

$$m\ddot{r(t)} = F(r(t)) \tag{2.15}$$

Where $F(r(t))$ is the forces acting on the atom, $\ddot{r}(t)$ is the acceleration of the atom and m is the atomic mass. By these equations one can derive the Velocity Verlet Algorithm.

One can write the time dependent differentiations as:

$$\dot{r}(t) = \nu(t)$$

$$\ddot{r}_i = \dot{\nu}(t), \quad \dot{\nu}(t) = \frac{F(r(t))}{m}$$

By writing $r(t + \delta t)$ as a Taylor expansion:

$$r(t+h) = r(t) + h\dot{r}(t) + \frac{h^2}{2}\ddot{r}(t) + \mathcal{O}(h^3) \tag{2.16}$$

and rewrite it as:

$$r(t+h) = r(t) + h\nu(t) + \frac{h^2}{2}\frac{F(r(t))}{m} + \mathcal{O}(h^3) \tag{2.17}$$

Expanding $\nu(t+h)$ in the following step:

$$\nu(t+h) = \nu(t) + h\dot{\nu}(t) + \frac{h^2}{2}\ddot{\nu}(t) + \mathcal{O}(h^3) \tag{2.18}$$

Knowing that $\frac{F(r(t))}{m}$ will replace $\dot{\nu}(t)$ in the above equation, but an expression for $\ddot{\nu}(t)$ needs to be found. By using Taylor expansion for $\dot{\nu}(t+h)$, $\ddot{\nu}(t)$ can be expressed as:

$$\frac{h^2}{2}\ddot{\nu}(t) = h(\dot{\nu}(t+h) - \dot{\nu}(t)) + \mathcal{O}(h^3) \tag{2.19}$$

By using the above expression and the equation of motion in $\nu(t+h)$ the following result is obtained:

$$\nu(t+h) = \nu(t) + \frac{h}{2m}(F(r(t+h)) + F(r(t))) + \mathcal{O}(h^3) \tag{2.20}$$

The Velocity Verlet Algorithm has the following steps: Start to decide values for the position r_k,the velocity ν_k and the force acting on the atom $F(r_k)$. Then calculate r_{k+1} from $r(t+h)$, the next step is to evaluate $F(r_{k+1})$ and then decide ν_{k+1}. The first iteration is done, then start to repeat the procedure with starting values of the position, velocity and the force as r_{k+1},ν_{k+1} and $F(r_{k+1})$. [9]

Chapter 3

Methods

In this chapter the methods for comparing the different potentials provided by VASP are described; the code LAVAX that optimize the execution of VASP, as well as the methods to analyze the outputted data from VASP.

Since we have periodic boundary conditions in VASP we need a simulation cell large enough so that the cascade of high energetic atoms from the PKA don't cross over the boundary and interferes with itself (Fig. 3.1). A large cell can however be very costly to simulate.

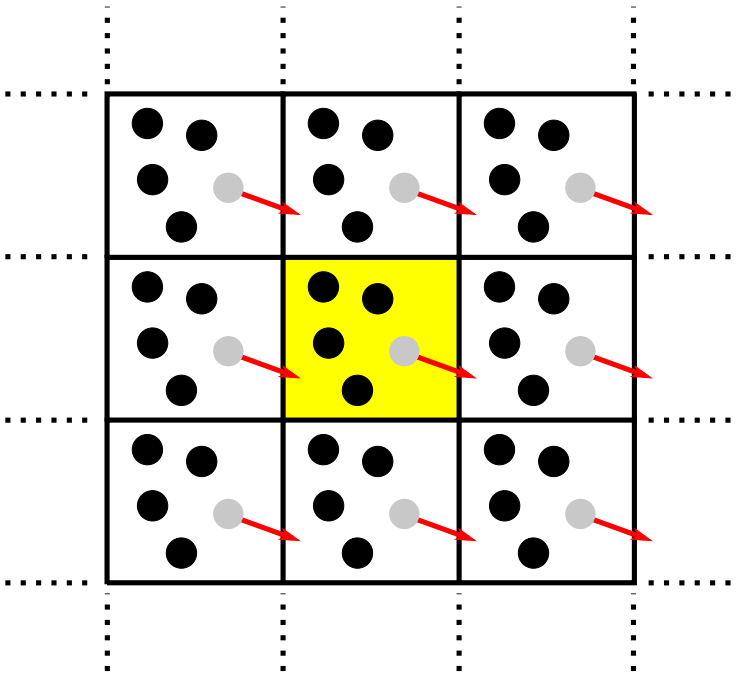


Figure 3.1: Periodic boundary conditions: Atoms from a collision cascade, crossing the boundary would (unrealistically) interfere with simulation cell.

3.1 Comparing Potentials

VASP provides a large number of PAW potentials. For some elements, several PAW versions exist, where the standard version generally has no extension. The extensions `_pv` and `_sv` imply that the p and s semi-core states are treated as valence states (i.e. for `V_pv` the $3p$ states are treated as valence states, and for `V_sv` the $3s$ and $3p$ states are treated as valence states). [7]

In this work we have used the potential `W` (tungsten) with the $6s$ and $5d$ orbitals treated as valence states with a total of 6 electrons; `W_pv` with the additional $5p$ orbital (a total of 12 electrons) and `W_sv` with the additional $5s$ (14 electrons total).

We need a way to determine at what distance the two pseudo-potentials start to deviate significantly from each other. This should become apparent when the atoms electron shells starts to overlap.

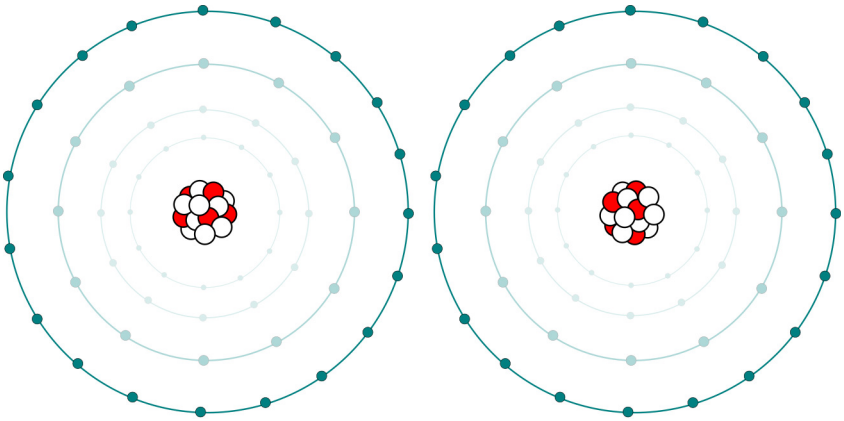


Figure 3.2: When the atoms are far enough apart, it is sufficient to only include the valence electrons in the potential.

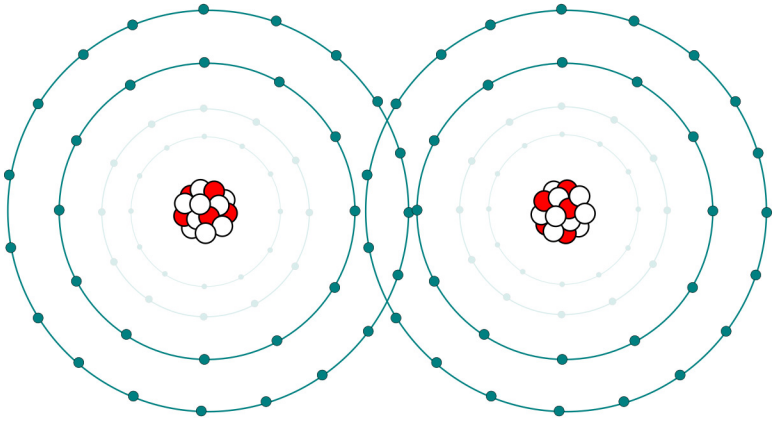


Figure 3.3: When the atoms are close enough so that their electron shells begin to overlap, it is necessary to include further shells in the potential.

We have two methods to compare the `W` and `W_pv` potentials. For both methods, the `NSW` flag in VASP is set to 0 so that we calculate the energy of the system statically. The first is a quasi-static movement of an atom as illustrated in Fig. 3.4. The rest of the atoms are kept at their respective lattice positions.

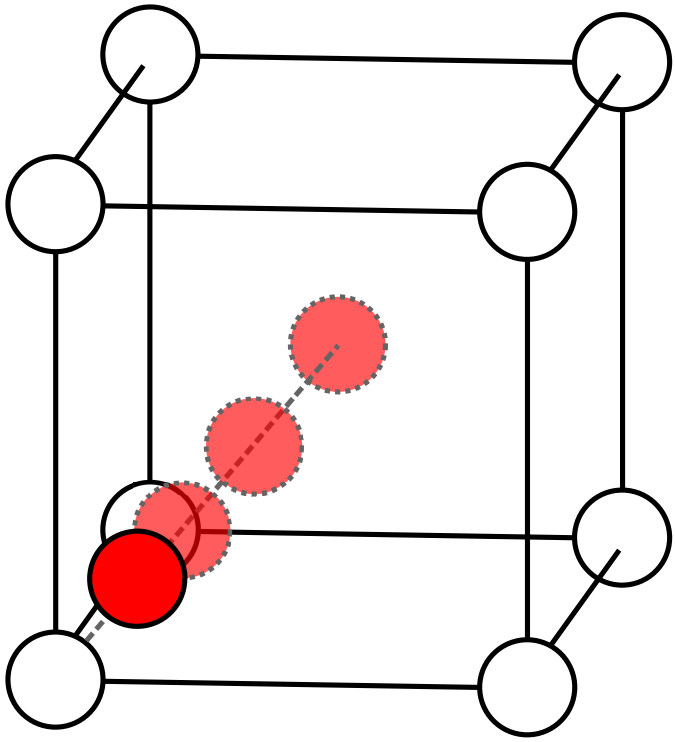


Figure 3.4: Determining the potential energy by a quasi-static method. One of the atoms is moved closer to another atom in each iteration and the energy of the system is computed.

The second method is compressing the entire lattice by varying the lattice constant as illustrated in Fig. 3.5. This will give us a zero-level if the lattice constant $a \to \infty$

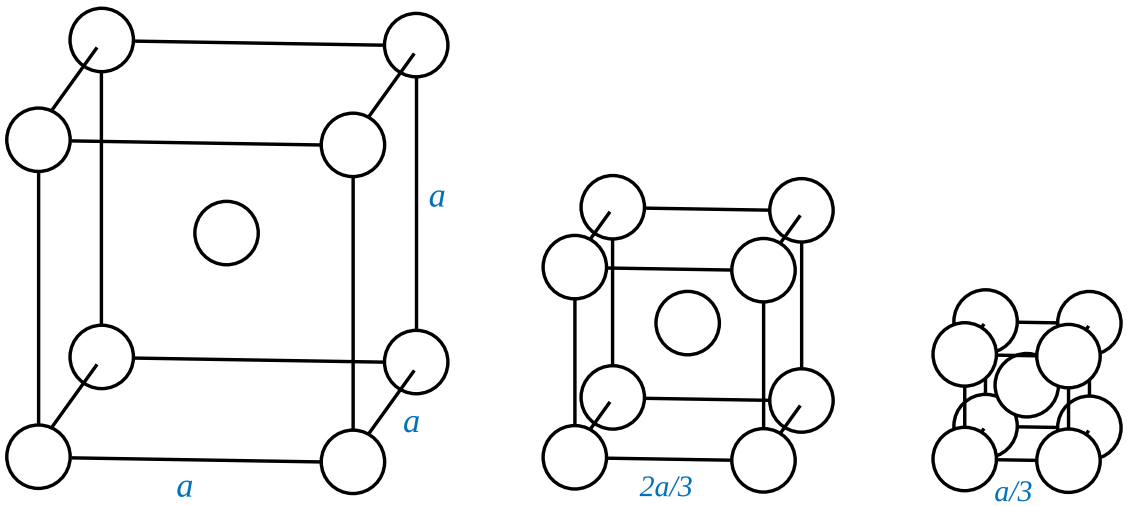


Figure 3.5: Determining the potential energy by varying the lattice constant. This can be thought of as compressing the entire crystal.

3.2 LAVAX - LAMMPS VASP Exchanger

We wrote a program (in java) called LAVAX that exchanges data between VASP and LAMMPS in order to speed up the execution time of a VASP simulation. LAMMPS is used to determine the set of atoms that will experience strong local compression conditions (see Fig. 3.6). The selected set of atoms will be treated in VASP by a potential that includes semi-core electrons in order to model the correct physics for these conditions. The remaining atoms can be treated with a minimal set of valence electrons to save computation time.

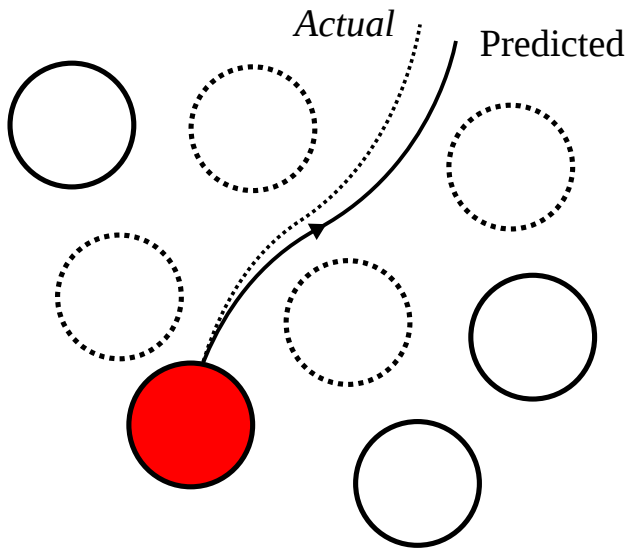


Figure 3.6: The predicted path in LAMMPS an atom may take v.s. the actual path it will take in VASP. The dashed atoms are those whose distance is short enough to warrant a more accurate potential during a run in VASP.

First, the state of the crystal (positions and velocities of the atoms) is imported to LAMMPS. LAMMPS then predicts the paths the atoms will take in the crystal during some short time step (on the order of 10 fs). LAMMPS will simultaneously build pairwise neighbor lists for atoms within some specified cutoff distance r_c of each other (determined perhaps from one of the methods in the previous section). VASP then runs for the same short time step, with the selected atoms from the neighbor lists using a more accurate potential model compared to the rest of the atoms in the crystal.

LAVAX assigns an index to each atom for bookkeeping and LAMMPS is configured to output the indices in the neighbor lists.

The predicted path differs from the actual path the atoms take in VASP (see Fig. 3.6), but since the time step is small, this difference is negligible. The process can be repeated for desired number of time steps.

The algorithm is described in Fig. 3.7.

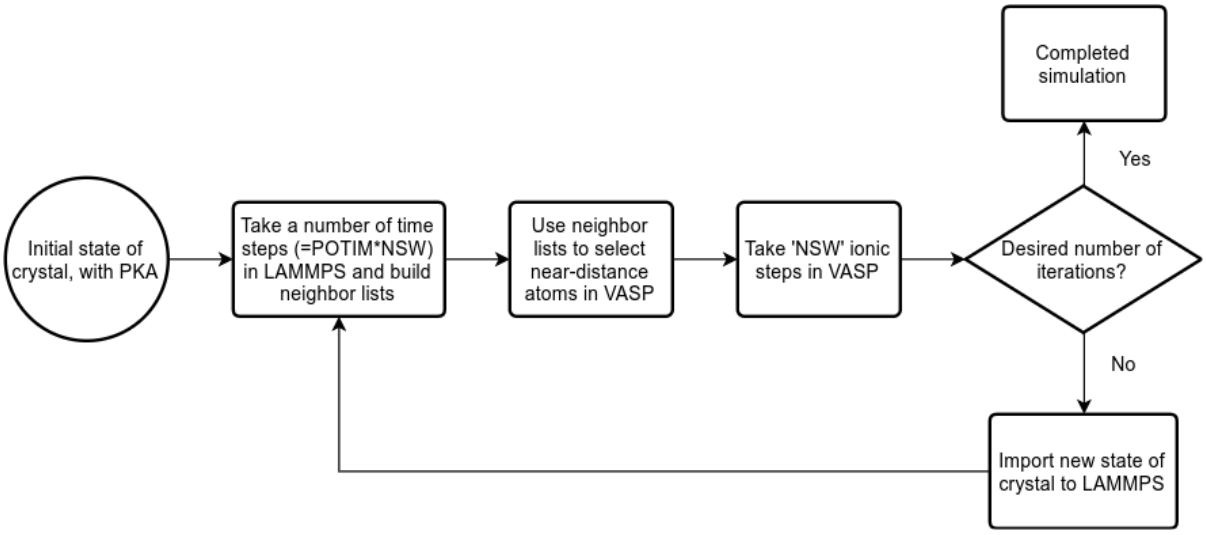


Figure 3.7: Principal flow chart of LAVAX.

3.2.1 Adaptive timestep

In the beginning of the simulation it can be of interest to take smaller time steps since a few of the atoms will have comparably high speeds (such as the PKA). Later in the simulation, the kinetic energy will distribute itself over the entire crystal and the maximum speed among all atoms will be much smaller. A larger timestep will then suffice. We can use an adaptive time step Δt chosen with the following condition in the beginning of each LAVAX iteration:

$$\Delta t = \min\left(\texttt{MAX\_DISTANCE}/v_{max},\ \texttt{MAX\_TIMESTEP}\right) \tag{3.1}$$

where v_{max} is the maximum speed among all atoms in the crystal, `MAX_DISTANCE` is the maximum tolerated distance any atom may travel during one ionic step and `MAX_TIMESTEP` is the maximum tolerated time step so that we don't select an unfeasibly large time step.

The adaptive time step was implemented in LAVAX and used in all the simulations.

3.3 Methods for the analysis

We can use to total kinetic energy of the crystal as a proxy variable for the state of the entire crystal. To see if we get the same accuracy with LAVAX, we first need to perform a reference simulations with only the more accurate potential model for the entire crystal (we can also perform one for the less accurate potential model). We then run the same simulation with LAVAX for various values for the r_c-cutoff parameter, where we expect the systems development to approach the development of the reference simulation for larger r_c-values.

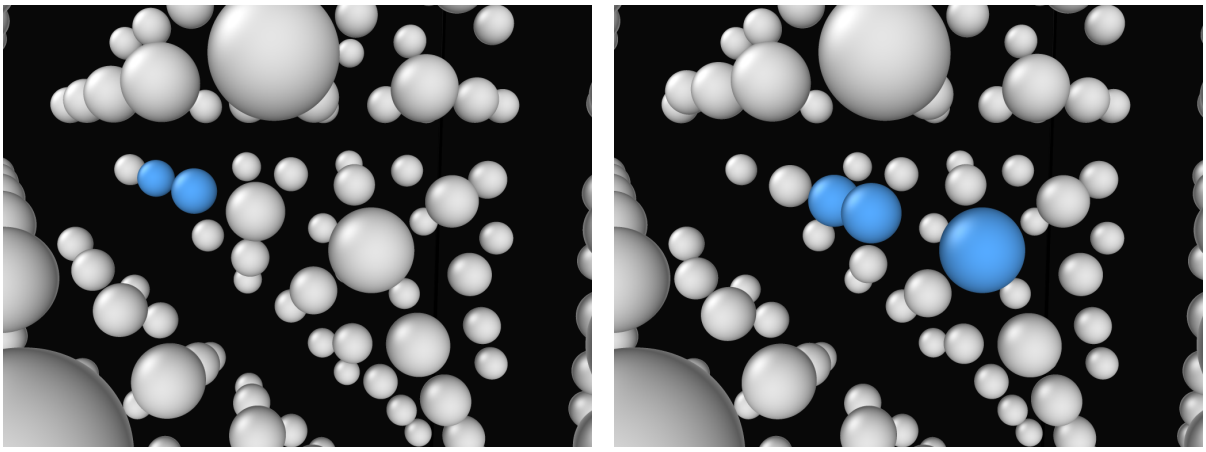
To determine the performance gain using LAVAX, we can compare the average execution time of all the electronic steps in VASP with that of the reference simulation. Since LAMMPS runs vastly faster than VASP we can neglect it's contribution to the total execution time.

We can also compare the total wall time of a full simulation, which is probably what a prospective user would be more interested in.

Chapter 4

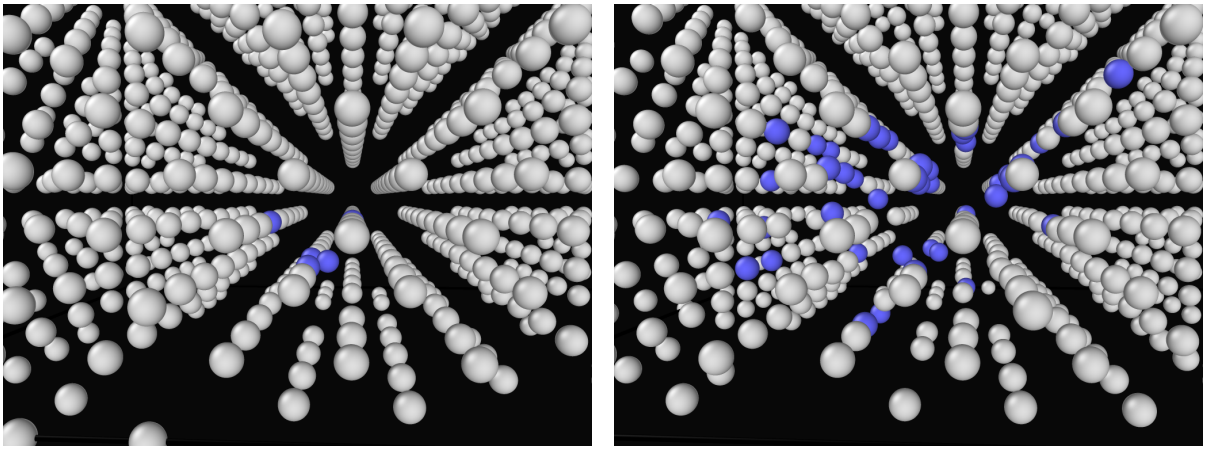
Results

Here we see a visualization of the switch of potentials during a LAVAX run.



(a) t_0 (b) t_1

Figure 4.1: A switch of potentials between `W` (grey) and `W_pv` (blue) during a LAVAX run



(a) t_0 (b) t_1

Figure 4.2: A switch of potentials during a LAVAX run in a larger crystal

4.1 Comparing Potentials

Using the method of quasi-static movement of one atom, the potentials `W` and `W_pv` (as well as `W_sv` even though it wasn't used in any of the full simulations) as a function of distance was determined. The divergence of the two can clearly be seen in Fig. 4.3

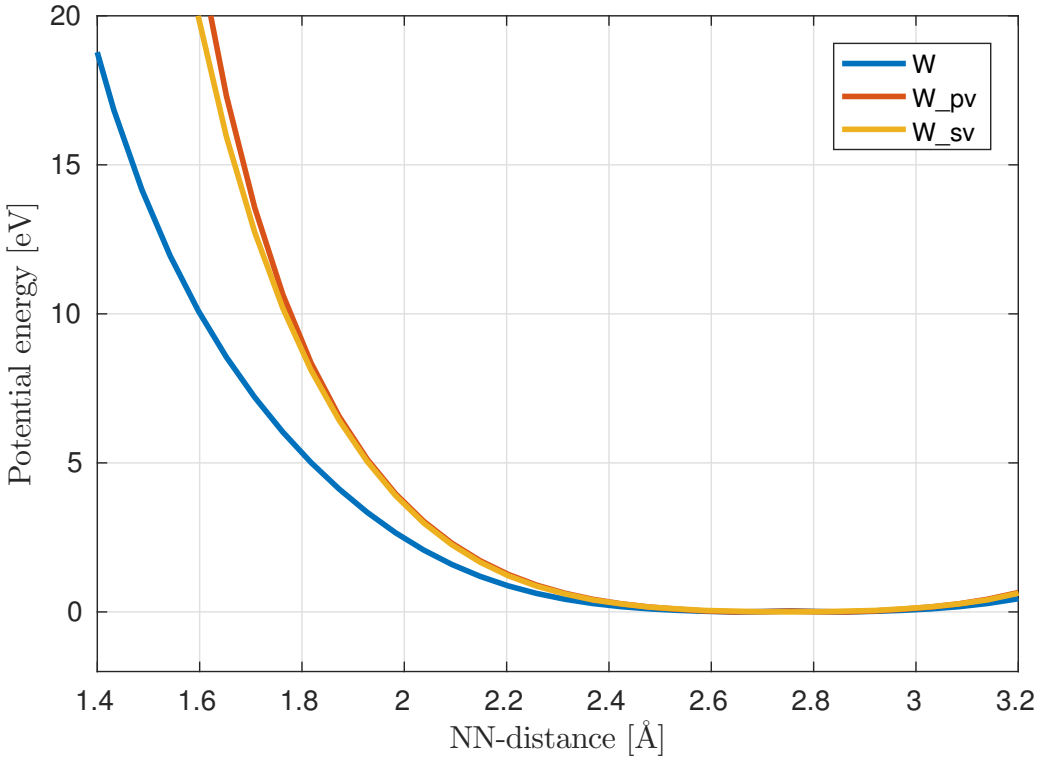


Figure 4.3: Comparing the potential energy using the quasi-static method.

When we instead use the method of varying the lattice constant, we get the results in Fig. 4.4. Here we have also inserted the theoretical screened Coulumbic potential (ZBL) for comparison.

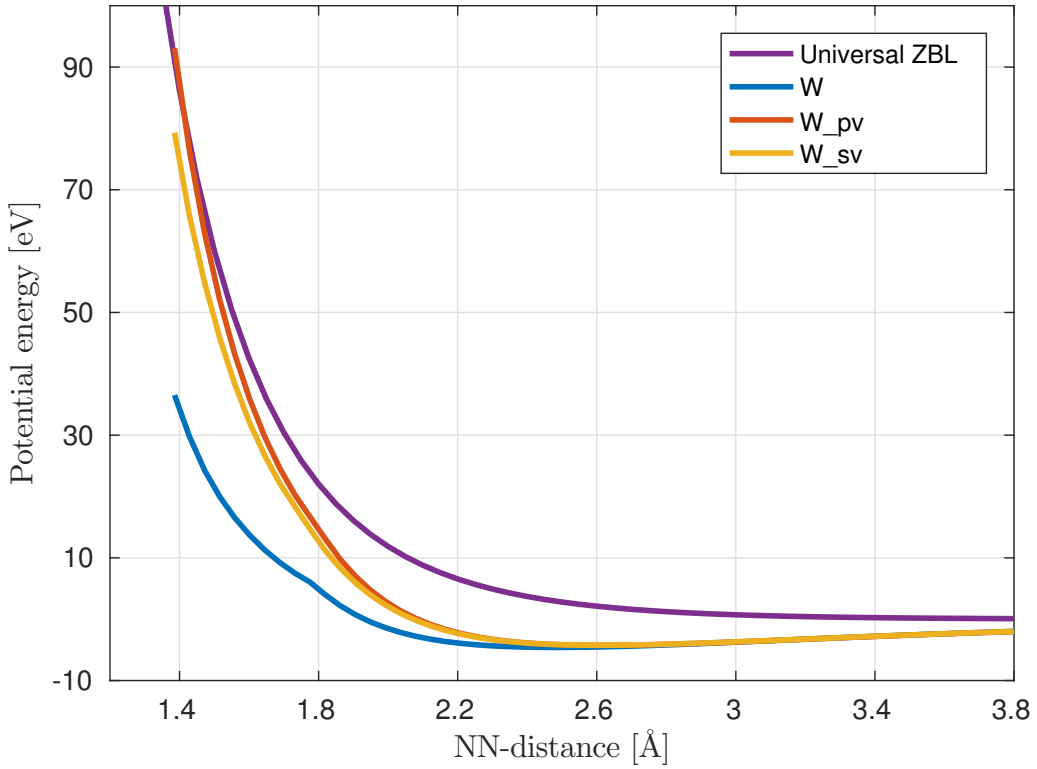


Figure 4.4: Comparing the two potential models by varying the lattice constant.

From both Fig. 4.3 and 4.4 we can clearly see that the potential `W` becomes inadequate around the NN-distance $\approx$ 2.4 Å. We also see that the more refined `W_pv` (and `W_sv`) approaches the ZBL-curve at short distances, as expected.

4.2 Kinetic energy during a LAVAX run

First, we ran VASP simulations with only the `W` and `W_pv` for the entire crystal, respectively, for reference. The system studied was a bcc crystal consisting of 648 ($= 2\times9\times6\times6$) W-atoms, with initial temperature 50 K. A PKA was given a kinetic energy of 100 eV, in the direction $\langle 135 \rangle$. We ran ten simulations for each potential. The total kinetic energy of the system is plotted in Fig. 4.5 and can be seen as a proxy variable for the systems state.

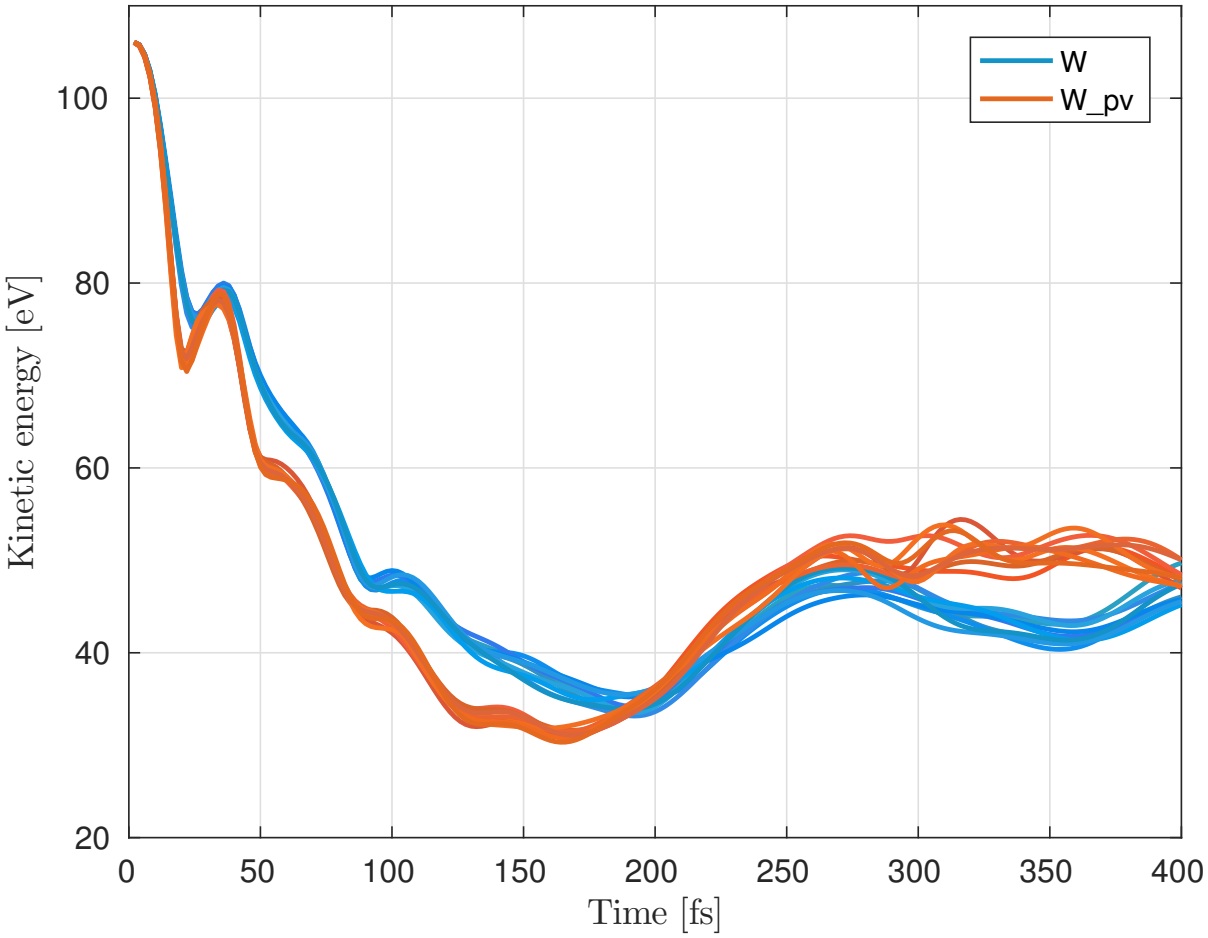


Figure 4.5: The kinetic energy of the system with `W` and `W_pv` for ten runs each with random initial velocities, from the Maxwell-Boltzmann distribution.

Next, we ran the LAVAX program with different values for the r_c-cutoff. That is, the distance in which the `W_pv` potential should be used. 10 runs for each r_c-value was started, although most of them diverged before the simulation could finish. Because of this, not a very good statistical base was attained. A few more simulations was run at a later date to attain a better base to analyze.

The mean value of the entire systems kinetic energy is given in Fig. 4.6

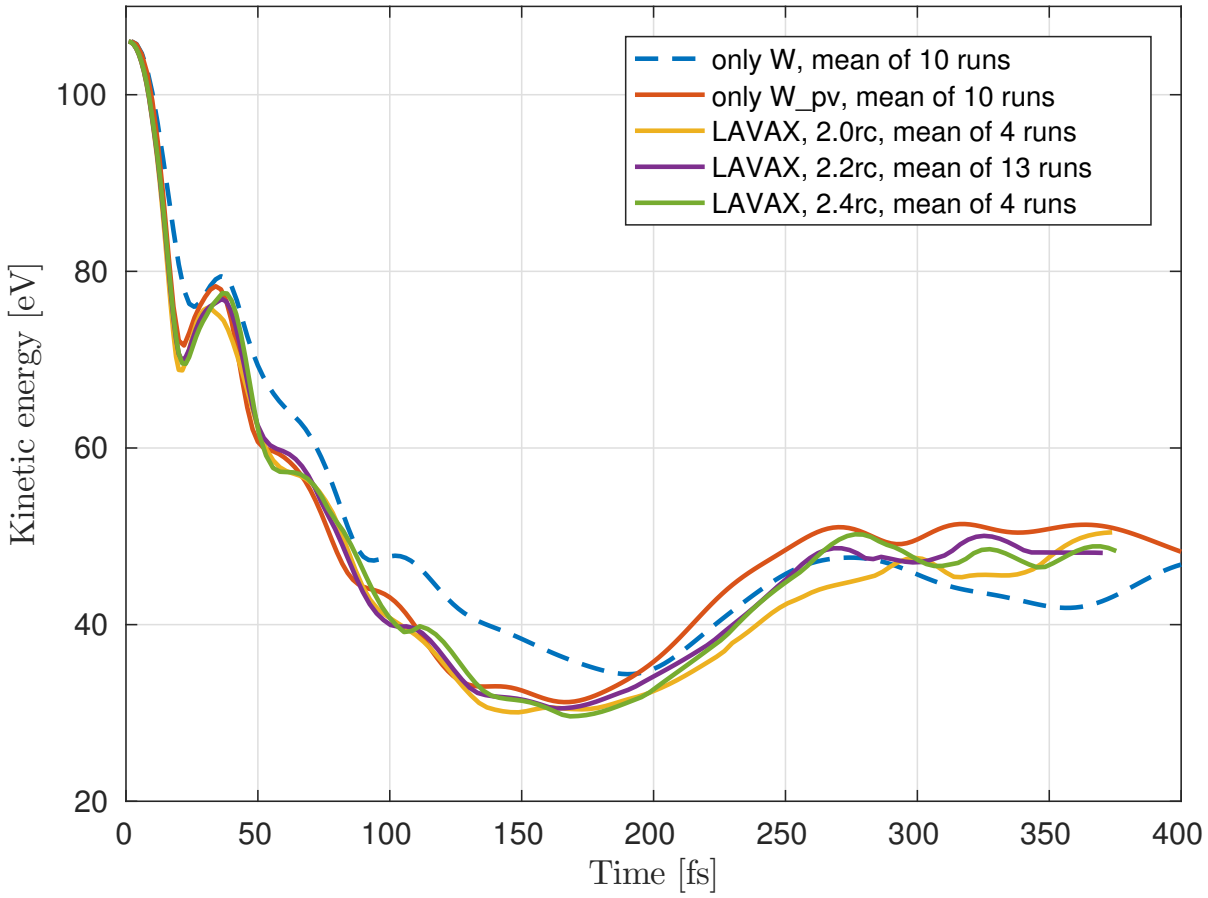


Figure 4.6: The mean value of the systems kinetic energy, with random initial velocities from the M-B distribution.

From Fig. 4.6 we see that the systems development initially agrees well for all r_c-values, but begins to depart noticeably around 200 fs. One must remember that the system is chaotic in nature: a small change in velocity, position or potential energy at an early stage, eventually leads to a large change at a later time. This is why it is important to have a large statistical base, as the mean value of many runs should correspond the systems "real" development.

A zoom-in of Fig. 4.6 can be seen in Fig. 4.7

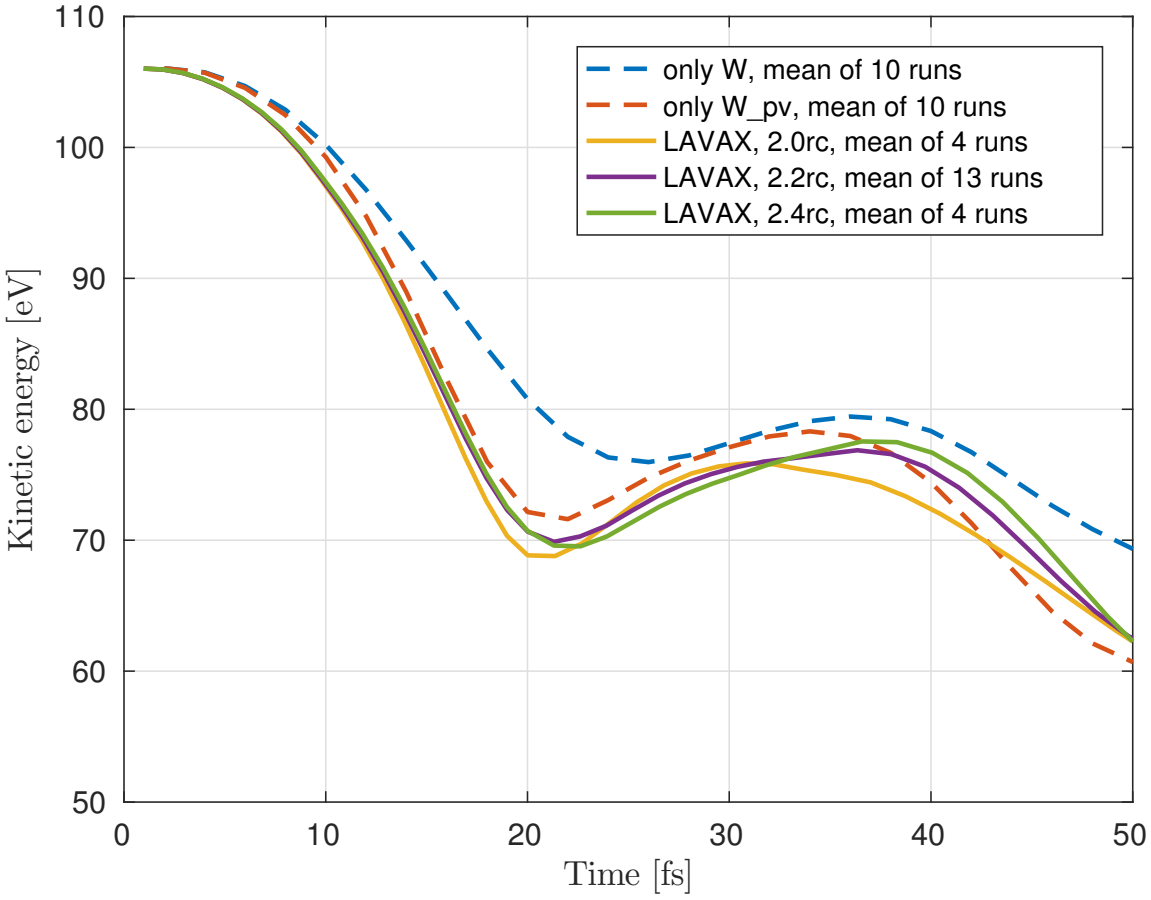


Figure 4.7: The mean value of the systems kinetic energy.

Here, we get a better view of the systems development in the beginning. The LAVAX runs are clearly closer to the `W_pv` reference simulation than the `W`, even though we don't get *perfect* conformity to the `W_pv`. Whether this difference is acceptable or not is difficult to say.

4.3 Performance gain with LAVAX

We ran a VASP simulation through LAVAX, as well as reference simulations, on the Marconi supercomputer with the following settings:

`CPU Cores:` 2400
`W crystal atoms:` 1980 $(= 2 \times 9 \times 10 \times 11)$
r_c`-cutoff:` 2.2 `[Å]`
`PKA energy:` 300 `[eV]`
`PKA direction:` $\langle 135 \rangle$
`---- Adaptive time step settings ----`
`MAX_DISTANCE: 0.1 [Å]`
`MAX_TIMESTEP: 3.0 [fs]`

The average execution time spent in one electronic loop (every ionic step requires about 10 to 40 of these) is presented in Fig. 4.8

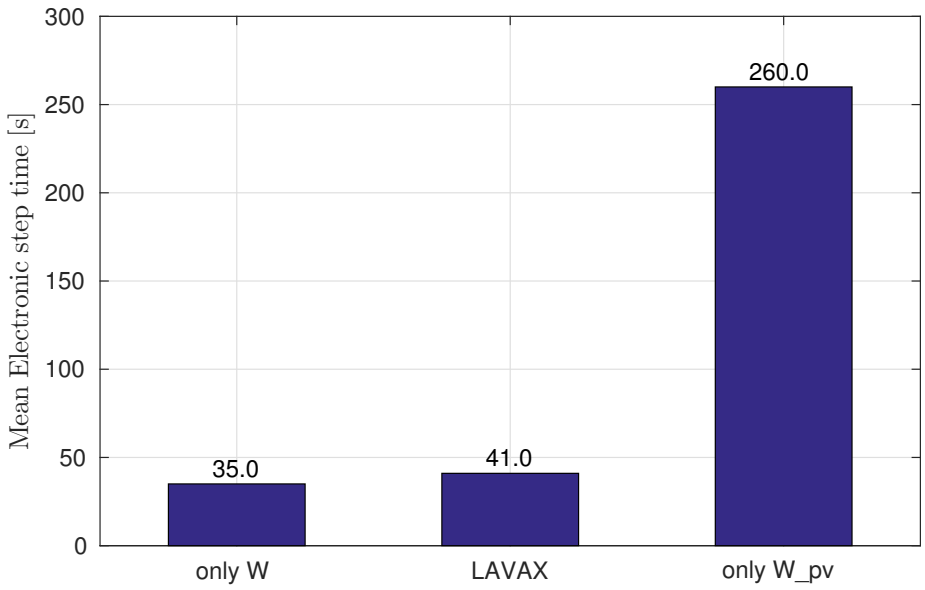


Figure 4.8: The average time spent in one electronic step for LAVAX.

Since the simulation ran on 2400 cores, the `W_pv` spent 2400×260 s ≈ 173 core hours for an average electronic step.

We ran another, smaller VASP simulation via LAVAX (plus reference simulations) on the Marconi supercomputer with the following settings:

`CPU Cores:` 1600
`W crystal atoms:` 648 $(= 2 \times 9 \times 6 \times 6)$
r_c`-cutoff:` 2.2 `[Å]`
`PKA energy:` 100 `[eV]`
`PKA direction:` $\langle 135 \rangle$
`Total ionic steps (NSW): 150`
`---- Adaptive time step settings ----`
`MAX_DISTANCE: 0.1 [Å]`
`MAX_TIMESTEP: 3.0 [fs]`

The total elapsed wall time for the entire simulation is shown in Fig. 4.9

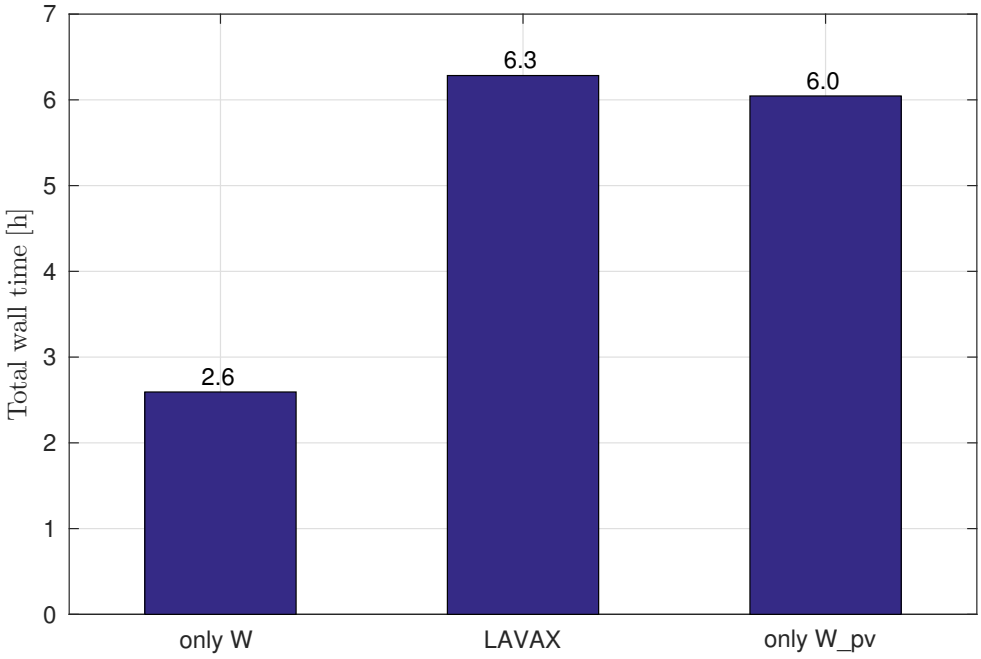


Figure 4.9: Comparison of the total elapsed wall time for a full simulation using LAVAX.

Chapter 5

Discussion

Studying the two potential comparison methods in Fig. 4.3 and Fig. 4.4 we see that we arrive at the same result for determining the r_c-cutoff value where the potentials begin to depart (≈ 2.4 Å). Any of the methods may be used for this task, although the quasi-static method seems more realistic than compressing the entire crystal, because not all atoms in the crystal are experiencing strong local compression conditions at the same time.

From the kinetic energy in Fig. 4.5, we see that the development of the crystals state is chaotic. This is why we need to take the average of many simulations with different initial velocities, as displayed in Fig. 4.6, in order to appreciate how much accuracy is lost when using LAVAX for different r_c-values.

We see that the performance gain is quite significant when comparing the execution time in Fig. 4.8 for LAVAX and only `W_pv` for the entire crystal. They differ by a factor 6.3!

If we assume that the time complexity for the number of electrons N scales as $\mathcal{O}(N^3)$, we can estimate how much larger a crystal we could simulate with the same amount of resources with LAVAX. We introduce the average number of valence state electrons n_{avg} for all atoms M in a crystal during a LAVAX run. The `W_pv` has 12 valence state electrons so we can write the performance gain as

$$\frac{(12M)^3}{(n_{\text{avg}}M)^3} = 6.3 \quad \Rightarrow n_{\text{avg}} = \frac{12}{6.3^{1/3}} \approx 6.5$$

If we want to simulate a crystal larger by a factor k in the same time it would take using `W_pv` for a crystal of size M we get

$$\frac{(12M)^3}{(n_{\text{avg}}kM)^3} = 1 \quad \Rightarrow k = \frac{12}{n_{\text{avg}}} \approx 1.85$$

It may seem tempting to conclude that the performance is worse with LAVAX when comparing the total elapsed wall time in Fig. 4.9. But the settings on the supercomputer may have been vastly different between different sets of runs: different process priorities and difference in performance between nodes may effect the resulting wall time. The figure does however give us some idea of how expensive these types of simulations are.

We have not run any test to determine the performance gain of the adaptive time step alone. The adaptive time step could although explain the difference in behavior of the systems in Fig. 4.7. The LAVAX simulations had a time step (`POTIM`) of about 1 fs in the beginning, while the `W_pv` used a constant 2.0 fs for the entire simulation. The LAVAX curves are therefor a bit smoother and could explain why the initial dip in kinetic energy goes deeper for LAVAX than the `W_pv` curve. This is, again, an unfortunate inconsistency of settings. The dip do however occur at the same time for LAVAX and `W_pv`, but not for `W` which lags behind.

It is unfortunate that VASP often diverge when you switch potential for many atoms before a new iteration.

VASP saves the calculated wave function in `WAVECAR` and it is used as the initial guess for the next ionic step, since recalculating it from the beginning is very expensive (the `WAVECAR` file can be over 100 GB in size). Since we may have switched potential for a large number of atoms during a LAVAX iteration, the `WAVECAR` file may be inconsistent with the "new" system and the DFT calculation could diverge as a result.

It is interesting to note that most of the LAVAX runs that were able to finish used $r_c = 2.2$, while a smaller $r_c = 2.0$ and larger $r_c = 2.4$ caused VASP to diverge more often. This could be explained that when using a smaller r_c value, the difference in potential energy between `W` and `W_pv` for that r_c is so large that when we switch potentials in a new iteration the `WAVECAR` becomes inconsistent. For larger r_c values we switch potential for a greater number of atoms and the `WAVECAR` becomes inconsistent for that reason. The sweet spot seems to be $r_c = 2.2$ for our simulations.

Chapter 6

Summary and Conclusions

The results from Fig. 4.8 show that LAVAX vastly improves the performance for radiation damage simulations, without sacrificing too much accuracy. For most performance critical applications, a mere 10 % increase in speed could be considered significant. Looking at the mean electronic step time, we got an improvement of about 500 %

From Fig. 4.9 we see that a typical 648 atom simulation finishes in about 6 hours on a supercomputer with 1600 CPU cores (a total of 9600 core hours). For perspective: running the same simulation on a single core machine would take over a year, which shows how computationally expensive these simulations are and how important any performance gain could be.

It is unfortunate that we don't have any fair comparison of the total wall time since the preconditions on the supercomputer may have varied. Hopefully, we would see equal scalability for the total wall time as for the mean electronic step time if we ran the simulations under the same settings.

An improvement of LAVAX would be to check if VASP diverges and if so remove `WAVECAR` so that VASP has to recalculate it from scratch when it is restarted. Removing `WAVECAR` every time to avoid divergence would be a bad idea since divergence only occurs occasionally and we would loose performance if we did.

All the energy of the PKA remain inside the simulation cell because of the periodic boundary conditions and the resulting temperature of the crystal will be very large. To quantify the remaining defects we would need to remove some of the energy at the boundaries, to model the spread of heat to the rest of the material (in a way freezing the simulation cell).

Right now, LAVAX only works for one particle type. To develop LAVAX in the future one could generalize the program by making it possible to use more than one particle types. Radiation damage in a material will with time develop the material into an alloy. By generalizing LAVAX one could simulate larger crystals of alloys to study the behavior of the material.

Appendix

LAVAX source code

The source code for LAVAX, as well as the code for the potential comparison, is available at

`https://github.com/danielk707/lavax.git`

Bibliography

[1] G. J. Ackland, "Interatomic Potential Development", *Comprehensive Nuclear Materials* **1**, 287 (2012).

[2] J. P. Crocombette and F. Willaime, "*Ab Initio* Electronic Structure Calculations for Nuclear Materials", *Comprehensive Nuclear Materials* **1**, 224, 226 (2012).

[3] S. J. Zinkle, "Radiation-Induced Effects on Microstructure", *Comprehensive Nuclear Materials* **1**, 65-93 (2012).

[4] G. Pintsuk, "Tungsten as a Plasma-Facing Material", *Comprehensive Nuclear Materials* **4**, 551-581 (2012).

[5] Antoine Claisse, "Ab Initio based Multi-Scale Simulations of Oxide Dispersion Strengthened Steels", 5-6; 8-9; 11-14; 20-21, MSc thesis, KTH, TRITA-FYS 2012;10

[6] `https://commons.wikimedia.org/wiki/File:Sketch_Pseudopotentials.png`

[7] "VASP the Guide, The PAW potentials", `http://cms.mpi.univie.ac.at/vasp/vasp/PAW_potentials.html`

[8] "The LAMMPS Manual", `http://lammps.sandia.gov/doc/Section_intro.html`

[9] Christian Leo Wennberg, "Exploring the Interactive Landscape of Lipid Bilayers", Licentiate Thesis, 5-6, KTH, TRITA-FYS 2014;24

[10] Lars Bergqvist, "Moderna beräkningsmetoder inom elektronstruktur och magnetism, presentation", *IM2602 Fasta tillståndets fysik, tilläggskurs*

[11] W. Koch and M. C. Holthausen, "A Chemist's Guide to Density Functional Theory, 2nd ed.", 6-8; 33-36; 44-45, (2001)

[12] S. Plimpton, "Fast Parallel Algorithms for Short-Range Molecular Dynamics", *Journal of Computational Physics* **117**, 1-19 (1995).

[13] W. Kohn and L. J. Sham, "Self-Consistent Equations Including Exchange and Correlation Effects", *Physical Review* **140** (4A), A1133-A1138 (1965).

[14] P. Blöchl, C. Först and J. Schimpl, "Projector augmented wave method: ab initio molecular dynamics with full wave functions", *Bulletin of Materials Science* **26** (1), 33-41 (2003).

[15] G. Kresse and J. Hafner, "Ab initio molecular dynamics for liquid metals", *Physical Review B* **47** (1), 558-561 (1993).

[16] G. Kresse "Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set", *Computational Materials Science* **6**, 15-50 (1996)

[17] J. P. Perdew, K. Burke and M. Ernzerhof "Generalized Gradient Approximation Made Simple", *Physical Review Letters* **77** (18), 3865-3868 (1996)