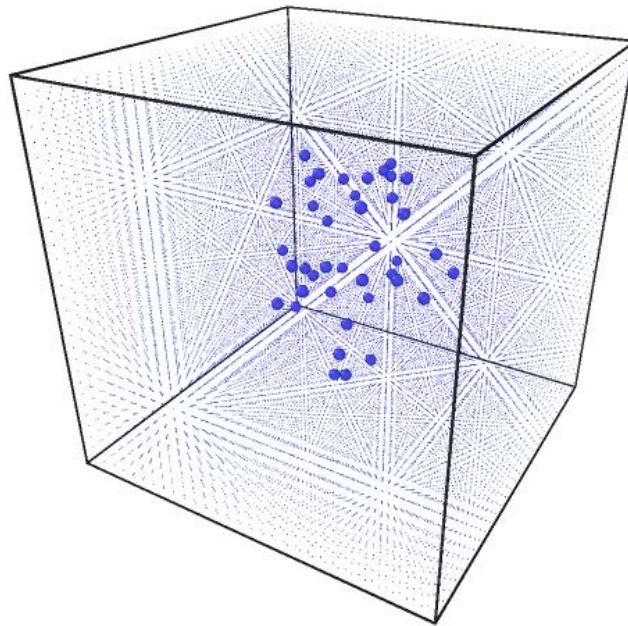


Investigation of H-center diffusion in sodium chloride by molecular dynamics simulation



by Karun Datadien

18-7-2013

Supervisors:

M.Sc. V. Veligura
Dipl.-Ing. Dr. G. Hlawacek

Committee:

M.Sc. V. Veligura
Dipl.-Ing. Dr. G. Hlawacek
Dr. E.S. Kooij
Dr. ir. W.K. den Otter

Abstract

Irradiation of a NaCl crystal with ions leads to the creation of several types of color centers, including F- and H-centers. Since opposed to F-centers, H-centers are mobile at room temperature, their behavior determines the crystal coloration process. Therefore we investigate clustering and migration of H-centers in a NaCl crystal on a 200ps timescale using a molecular dynamics simulation. The clustering is measured and mostly occurs in the first 20 ps of the simulation. The clusters are not seen to migrate through the lattice. Non-clustered H-centers are mobile, but do not seem to have a clear directional diffusion on this timescale.

Table of contents

Chapter 1: Introduction.....	1
Chapter 2: Theory	2
2.1 Color centers in alkali halides.....	2
2.2 Interatomic pair potentials.....	4
Chapter 3: Software and writing LAMMPS input scripts.....	5
Chapter 4: Preliminary Simulations.....	11
4.1 Diatomic energy and separation of $\text{Na}^+ \text{Cl}^-$	11
4.2 Cohesive energy of NaCl.....	13
4.3 Stress-strain curve of a cubic NaCl crystal.....	14
4.4 Li^+ bombardment on NaCl.....	16
Chapter 5: Main simulation: H-center clustering and cluster migration in NaCl	17
5.1 Simulation method	17
5.2 Potentials	18
5.3 Results	20
Chapter 6: Discussion & Conclusion.....	23
Chapter 7: References.....	25
Acknowledgements.....	26
Appendix A: Diatomic energy and separation of $\text{Na}^+ \text{Cl}^-$ scripts.....	27
Appendix B: Cohesive energy of NaCl script.....	29
Appendix C: Stress-strain curve of a cubic NaCl crystal scripts.....	31
Appendix D: Li^+ bombardment on NaCl scripts.....	34
Appendix E: Main simulation + clustering scripts.....	36

1. Introduction

According to the initial assignment the annealing of color centers at increased temperatures should have been studied in an experimental setup. Due to the problems with the experimental setup — in particular the ion-gun needed to create the color centers — after three weeks of experimental work the project was switched to the theoretical simulations. Molecular Dynamics (MD) simulations were performed in order to investigate the behavior of H-centers in sodium chloride (NaCl). This work should be seen as an exploration on the feasibility of MD simulations for the investigation of color center behavior in alkali halides.

When a NaCl crystal is exposed to ionizing radiation, pairs of Frenkel defects are created. The two parts of the pair are called a F- and H- center. These are the simplest type of color centers. The balance between these defects determines the crystal coloration after ion/electron or photon bombardment (of high enough energy), which was observed in ionoluminescence studies using a Helium Ion Microscope (HIM). To understand the experimentally obtained results, in the present work diffusion of H-centers is simulated at a timescale of several ps and a length scale of several nm using a molecular dynamics simulation to gain better understanding of their behavior. The amount of clusters formed, their center of mass and the number of atoms in each cluster are tracked over a timespan of 200 ps. This is done for two scenarios: one with a lower initial concentration of Cl atoms and one with a higher concentration, $0.00013 \text{ atom}/\text{\AA}^3$ and $0.00039 \text{ atom}/\text{\AA}^3$ respectively. This main simulation is described in Chapter 5.

In an effort to confirm the validity of the interatomic potentials that are used, several preliminary simulations were created (Chapter 4). Three separate simulations are used for finding the diatomic potential energy of Na^+ and Cl^- , the cohesive energy of NaCl and the stress-strain relationship of a cubic NaCl crystal at 100 K, 300 K and 500 K. The results are then compared with experimental results and calculations from several literature sources. Simulation of the ion bombardment of a NaCl crystal (Chapter 4.4) was performed using a different simulation method. Li^+ ions with energy of 100 eV and 200 eV were used in that simulation. We obtained sputter yields of the different species and compared these results with simulation results obtained from SRIM 2013.^[1]

Since it is likely that this work will be altered or expanded upon by others with little experience in MD, an introduction to writing an input script is presented in Chapter 3.

2. Theory

2.1 Color centers in alkali halides

NaCl is an alkali halide crystal that is often used for the studies of color centers. Alkali halides are useful for this purpose due to their optical properties — in particular their transparency over a wide wavelength range; they are transparent from the far ultraviolet into the far infrared. The UV absorption corresponds to electronic transitions, while the infrared absorption is related to small energy differences in the possible vibrational and rotational states of the ions composing the solid.^[2] The crystal structure of NaCl is that of two separate interlocked face-centered cubic lattices. Na⁺ (alkali metal ion - cation) is situated on the points of one of these lattices, Cl⁻ (halogen ion - anion) on the other.

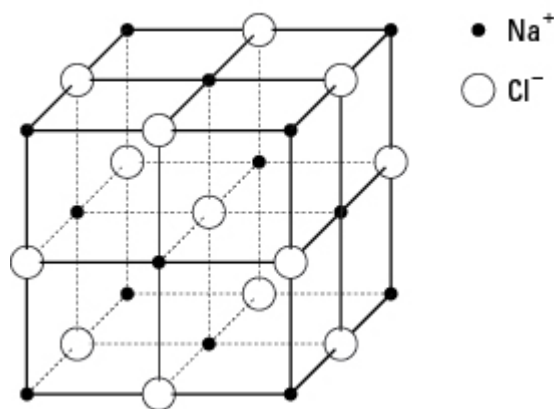


Fig. 1 NaCl crystal structure

Some alkali halides - CsCl for example - have a different structure, namely that of two interlocked simple cubic lattices. Changes in optical absorption are easily detectable due to the absence of spectral absorption lines in an alkali halide containing no, or very little, impurities.

Electrons and holes can be released in the crystal by some process (for example x-ray irradiation or electron bombardment) by stripping electrons from normal ions; forming an equal amount of electrons and holes. Most will recombine quickly reconstituting the normal ions.^[2] If before reconstitution the electron or hole wanders into an anion- or cation-vacancy respectively, it can be trapped in the defect's electric field. Thereby creating certain types of color centers. Note that the vacancies by themselves are also color centers called α -centers.

The F-center is the most common and best understood type of color center. When an electron is trapped at a positively charged anion vacancy, the resulting neutral center is called an F-center, from Farbzentrum (German for color center). This can be seen as a system analogous to a hydrogen atom. As such, it is expected to find several lines in the absorption spectrum in the visible region (considered to be wavelengths of 400nm to 700nm)

when F-centers are present in an alkali halide. This is the cause of coloration that can occur in alkali halide crystals.

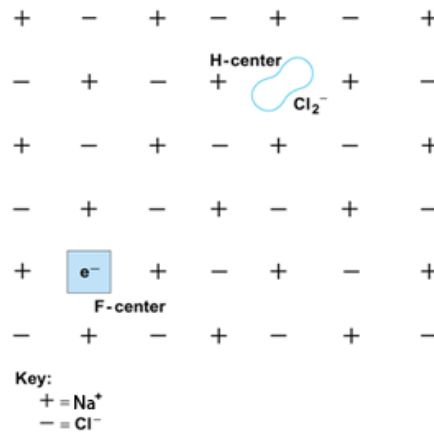


Fig. 2 F- and H- center in NaCl ^[3]

H-centers occur together with F-centers in a crystal (see Fig. 2). Due to impact or irradiation of NaCl an electron can be stripped from a Cl⁻ ion, leaving the electron at the vacancy site (this is a F-center) and putting the neutral Cl atom at an interstitial site near a halide ion (this is a H-center). H-centers are known to be mobile at temperatures above 80 K and may cluster.^[4] The diffusion mechanism of Cl atoms is as follows: the Cl atoms form a bond with lattice Cl⁻, creating a Cl₂⁻ molecule. The Cl₂⁻ molecule subsequently dissociates giving a Cl⁻ and neutral Cl atom. The neutral Cl bonds with a different lattice Cl⁻ ion, creating a new Cl₂⁻ molecule. This mechanism allows the Cl atoms to migrate through the crystal.

2.2 Interatomic pair potentials

For accurate results in MD simulations it is critical that the interatomic potentials are suitable and accurate. For example, there is a significant difference in the potentials used in solvated NaCl surrounded by water as opposed to solid NaCl in a crystal lattice. Also, certain interatomic potentials are only valid in a limited temperature range.

A wide array of types of pair potentials can be used in MD software — in this work a short-range Buckingham potential supplemented with a long range interatomic Coulomb potential was used.

$$U(r) = \frac{Z_1 \cdot Z_2 \cdot e^2}{r} \cdot D + A \cdot \exp\left(-\frac{r}{B}\right) - \frac{C}{r^6}, \quad r < r_c \quad (1)$$

A , B and C are constants and r_c is the cutoff range — set to 8.0 Å in all simulations in this paper. This value of r_c is chosen to decrease computation time substantially while still giving accurate results. Z_1 and Z_2 are the charges of each of the atoms/ions in the pair and e is the elementary electron charge. D is an energy-conversion constant. The first term describes the Coulombic electrostatic interactions, the second and third terms together represent the Buckingham potential. The second term on the right-hand side describes the repulsion due to overlap of the closed shell electron clouds. The third term describes the attraction due to the Van der Waals force between two atoms which are not directly bonded. It is clear that the second term is repulsive and the third term is attractive by taking the first derivative with respect to r .

The Buckingham potential is chosen because it gives a better description of the strong repulsion due to the overlap of the electron clouds when compared to some other potentials, such as the widely used Lennard-Jones potential. This can be especially important in simulations of bombardment by energetic atoms or ions^[5]. The improved accuracy has the drawback that the computing time is increased by approximately a factor of 4 compared to using the Lennard-Jones potential.

The potentials used in this work have been found in literature. They were found both in ready to use Buckingham form and in other forms, such as the Morse- and Moliere potential form. Wherever necessary these potentials were reparametrized to Buckingham parameters (more on this in Chapter 5). Most of these were then checked in several simulations to verify their validity/accuracy as described in Chapter 4. The parameters used in all simulations are those seen in Table 1, unless otherwise specified.

Interactions	A (eV)	B (Å)	C (eV Å ⁶)
Na ⁺ -Na ⁺	7895.4	0.1709	29.06
Na ⁺ -Cl ⁻	2314.7	0.2903	0.00
Cl ⁻ -Cl ⁻	1227.2	0.3214	29.06
$Z_{\text{Na}^+} = +0.988 \quad Z_{\text{Cl}^-} = -0.988$			

Table 1. Potentials from Ref. 6

3. Software and writing LAMMPS input scripts

There are many software packages available to create molecular dynamics simulations. For the simulations in this paper LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) was used^[2]. LAMMPS uses an input script – a basic text file – and can create dump files according to the user’s settings. These dump files – also basic text files – can be interpreted by visualization software such as OVITO (Open Visualization Tool)^[7] which is used to render all simulation images shown in this paper. The data from the dump files can be processed to get particular information in a particular form. Matlab was used for this purpose. Notepad++ was used to write the code.

In this section an overview is given of how LAMMPS input files are written. The purpose of this is to give a basic introduction for those not familiar with LAMMPS and also for the reader’s better understanding (?) of the input scripts given in the appendices. It may also be a good starting point for those wanting to expand or alter the simulations found in this paper. Below you will find the entire code of the main simulation (see Chapter 5) divided up into smaller parts, followed by a brief explanation and overview of some of the options. For more detailed information on all available options, it is recommended to look at the LAMMPS manual^[8].

Most scripts for the simulations described in this paper are divided into four sections: *initialization*, *atom definition*, *force fields* and *settings*.

1. Initialization

```
# H-center clustering in NaCl
# Karun Datadien | last update: 9-7-2013

# ----- INITIALIZATION -----
units          metal
dimension      3
boundary       p p p
atom_style     hybrid atomic charge
```

The units of the input values are defined here. In all simulations in this paper the command “metal” is used. This defines units such as eV for energy, Å for length, time in ps, temperature in Kelvin, etc. Next, the number of dimensions is chosen: the options are either 2 or 3.

The boundary type is then selected for planes perpendicular to the x-, y- and z-axis. There are three options: "s" for shrinkwrap, meaning the simulation box expands or shrinks to contain all atoms; "p" for periodic boundary conditions where there is interaction of atoms through the bounds of the simulation box; also we have "f" for fixed where atoms moving outside of the simulation box will be lost and the simulation will abort. If there are freely moving particles in the simulation it would therefore be wise to use a 'fix wall' command with this mode. This sets a repulsive force at the boundary.

An atom style must also be defined. There are many options and multiple styles can be defined by preceding their keywords by "hybrid". In this simulation two are chosen: namely "atomic" and "charge". This is appropriate for simulations with coarse-grain liquids, solids and/or metals where the atomic system also contains charges.

On computers with multiple processors, LAMMPS will partition the simulation domain into several 3D sub-domains, one of which is assigned to each processor. A "processor" command can be used to specify which processor will be assigned to a certain sub-domain. If this is not specified - as in this script - LAMMPS will automatically assign processors to sub-domains.

2. Atom definition

```
# ----- ATOM DEFINITION -----  
# Simulation box  
lattice      fcc 5.64 origin 0 0 0  
region      sample block 0 15.0 0 15.0 0 15.0  
region      DefectRegion sphere 7.5 7.5 7.5 4.5  
  
create_box  3 sample  
  
#Na+  
lattice      fcc 5.64 origin 0 0 0  
create_atoms 1 region sample  
  
#Cl-  
lattice      fcc 5.64 origin .5 0.5 0.5  
create_atoms 2 region sample  
  
#Cl (neutral)  
create_atoms 3 random 40 789465 DefectRegion  
group       NeutralClgroup type 3
```

First we define the type of lattice (fcc, bcc, hcp, etc.), the lattice constant a and origin in x-, y-, z-coordinates. One or multiple regions must be defined which will be filled with atoms later on. Following the "region" command we must define an ID - this can be any word or number - followed by the shape of the region such as block, sphere or cylinder. The start and end points on the x-, y- and z-axis are then set on the same line. The lattice must be defined

first, because the region's x-, y- and z-coordinates are set in number of lattice constants, not length units. A spherical region is also created with the origin at $[7.5, 7.5, 7.5]a$ and a radius of $4.5a$.

A simulation box is created next with "create_box" followed by the number of types of atoms that will be in the box followed by the region-ID. This way the box will assume the boundaries of this region.

Next we can fill the regions with atoms using "create_atoms". Na^+ and Cl^- will be placed according to the most recently defined lattice constant. At this point Na^+ has the ID "1" and Cl^- "2". "DefectRegion" is then filled with 40 randomly placed neutral Cl atoms. For random placement it makes no difference what kind of lattice options are set, the coordinates are completely random. We can choose the amount of atoms and must provide LAMMPS with a random seed, which can be any positive integer. Sometimes several different random seeds must be tried to prevent overlap or placement of atoms too close together if this is not desired. Everything else being equal, one particular random seed will always generate the same positions for the atoms. We define a group called "NeutralClgroup" containing all atoms of type "3", which are of course the neutral Cl atoms. This will be useful later on.

```
#Set masses
mass 1 22.98976928      #atomic weight sodium
mass 2 35.453          #atomic weight of chlorine-
mass 3 35.453          #atomic weight of neutral chlorine

#Set charges
set type 1 charge +0.988
set type 2 charge -0.988
set type 3 charge 0.0

#Misc.
neighbor 0.3 bin
neigh_modify delay 0
```

The masses and charges are set. These commands speak for themselves. Finally, there are two commands which provide LAMMPS with instructions on building neighbor lists. LAMMPS uses these neighbor lists to keep track of nearby particles, which enhances computational efficiency. The "Neighbor" command determines the way the pairwise neighbor lists are built. All atom pairs within a neighbor cutoff distance equal to their force cutoff (which will be defined in the *force fields* section) plus the so called skin distance - set at 0.3 in this script - are stored in the list. A larger skin distance usually means that neighbor lists need to be built less frequently, but more pairs must be checked for possible force interactions each timestep. Tweaking this value can optimize the computing efficiency. The "neigh_modify" command sets parameters that affect the frequency with which neighbor lists are built. The "delay" setting instructs LAMMPS to not build a new list until *at least* N steps after the previous build.

3. Force fields

```
# ----- FORCE FIELDS -----
pair_style          buck/coul/long 10.0 8.0
pair_coeff           1 1 7895.4 0.1709 29.06          #Na+ Na+
pair_coeff           2 2 1227.2 0.3214 29.06          #Cl- Cl-
pair_coeff           1 2 2314.7 0.2903 0.00           #Na+ Cl-

pair_coeff           3 3 -367.4337814 .5005465005 -275.0537438 #Cl Cl
pair_coeff           2 3 1205 .3358 0                 #Cl- Cl

pair_coeff           1 3 1012.6 .2903 0               #Na+ Cl

kspace_style        ewald 1.0E-3

#Temperature
group               samplegroup region sample
velocity            samplegroup create 300.0 43454 dist gaussian mom yes
```

First we define the pair style as Buckingham-Coulomb and set cutoff ranges for these pairwise interactions. The parameters for the Coulombic interaction are simply the charges set in the atom definition section. The parameters for the Buckingham interaction are set with “pair_coeff”. Perhaps the most laborious part of creating these simulations is finding the proper pair coefficients. These can either be calculated, found in literature or fit to experimental data. The three coefficients in this case are A , B and C as seen in (1).

The “kspace_style” command sets a long-range solver for LAMMPS to use each timestep. The name of this command comes from the fact that most long-range solvers perform their computation in K-space. Ewald summation is used here, which is one of the appropriate choices for “coul/long”. The lower the number following “ewald”, the more accurate the simulation becomes. However, it takes much more computing time. A good middle ground for most simulations is found to be 1.0E-3.

Finally, we set a temperature and pressure for the system. A group is defined to select all atoms in the *sample* region, which includes all atoms in the simulation in this case. We set the system to 300 K and instruct LAMMPS to perform time integration on Nosé-Hoover style non-Hamiltonian equations of motion, which will generate positions and velocities sampled from the isothermal-isobaric (npt) ensemble. The system’s temperature will oscillate a bit throughout the simulation and to minimize this, a damping coefficient is set (0.01). There are many options for setting velocity, temperature and pressure, all of which can be found in the LAMMPS manual.

4. Settings

```
# ----- SETTINGS -----  
compute peratom all pe/atom  
compute Clustercount NeutralClgroup cluster/atom 4  
compute 1 all temp  
  
# Dump cfg  
dump      1 all cfg 100 dump.NeutralClLowdensity_*.cfg id type xs ys zs fx fy fz c_peratom c_Clustercount  
dump_modify      1 pad 3 element Na Cl Cl_Neutral  
  
# Display thermo  
timestep      .005                                #.001 = femtosecond  
thermo_style  custom step pe pxx pyy pzz temp press  
thermo              50  
  
run              40000  
  
# ----SIMULATION DONE-----  
print "Simulation complete"
```

Here we instruct LAMMPS to compute certain data which we can then output to our dump file and eventually process with Matlab. In this case we compute the potential energy for each atom, then the clustering of neutral Cl atoms - with a cutoff range of 4Å - and finally the temperature of a group of atoms, which in this case is every atom in the simulation. On clustering: LAMMPS always assigns every single atom with a unique atom-ID. When an atom in *NeutralClgroup* is within the cutoff range of another atom in that group, they will be assigned a cluster-ID which will be set to the lowest numbered atom-ID in the cluster. All atoms outside of the *NeutralClgroup* get a cluster-ID of 0 assigned to them.

Next, we will dump a .cfg file which is an output file that can be interpreted by visualization software such as OVITO. These files can be opened and edited like any plain text file. A .cfg file is set to be dumped once every 100 timesteps with the “dump” command. Increasing the rate of creating dump files does not make a big difference concerning computing speed, but many dump files for a large system such as this (> 25000 atoms) can easily take up to a few GBs of disk space, since each dump file is about 2.5MB. In the dump file we write the x-, y- and z-coordinates (xs, ys, zs) of each atom, as well as the forces in x-, y- and z-direction acting on each atom (fx, fy, fz). We also dump the potential energy we computed earlier simply by including the command “c_<compute ID>”, specifically in this case “c_peratom”. The same is done for the computed clustercount with “c_Clustercount”.

Next we set the timestep to 0.005ps (5fs). The “thermo_style” will be displayed every 50 timesteps while the simulation is running. It will display the runtime in number of timesteps at that moment and certain components of the pressure tensor and the temperature and pressure of the system. Keeping an eye on these values is useful to check if the simulation

is doing what is expected, and there are no sudden temperature/pressure spikes or drops if this is not desired. In the end we set the number of timesteps to run the simulation for.

This has been a brief explanation to give a general idea of how a LAMMPS input file is written. As mentioned before there are many more options not discussed here, as evidenced by the over 1000 pages long LAMMPS manual. Beyond that, LAMMPS is an open-source program, so with knowledge of C++ the sourcecode can be expanded upon to include even more options. There are several research groups who have already done so and have made their work available online.

4. Preliminary Simulations

4.1 Diatomic energy and separation of $\text{Na}^+ \text{Cl}^-$

```
LAMMPS input:      Diatomicpotential.txt  (Appendix A)
Matlab script:     DPotentialgraph.m      (Appendix A)
Timestep: 0.02 fs | Runtime: 40 ps
```

The purpose of this simple simulation is to find the diatomic potential curve and compare it with experimentally acquired values. Na^+ and Cl^- ions are placed in the simulation box at a separation of $0.1a$, with $a = 5.64 \text{ \AA}$, immediately resulting in strong repulsion (Fig. 3). The separation distance and total potential energy are output every 10 timesteps. The solid blue line in Fig. 4 represents the obtained potential curve.

Since the exerted forces rapidly become large at short separation distances, a small timestep was chosen for the accuracy of the calculation. Note that the simulation box was deliberately chosen to be quite long ($40a$ in the x-direction), even though this increases computing time significantly because the processors are mapped over the entirety of the box, even if empty. The reason for this is the periodic boundary conditions, which is necessary for the Ewald summation, which in turn is necessary for determining the Coulombic interactions. With periodic boundary conditions the simulation box is three-dimensionally tiled. When an ion in the simulation box is within interaction cutoff range of another ion in one of the surrounding tiles, they will interact which is unwanted. We prevent that by making the box longer. Note also that the interatomic potential for a diatomic system is quite different from the potential energy per ion pair in a bulk crystal (see next simulation).

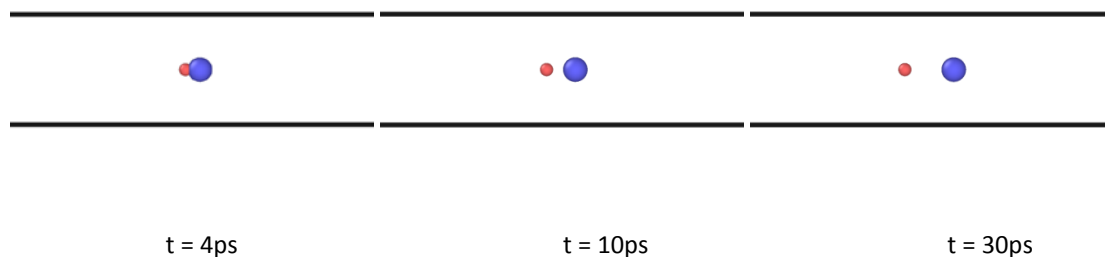


Fig. 3. Visualization of three states of the *diatomic energy and separation of $\text{Na}^+ \text{Cl}^-$* simulation.

Results and comparison with literature

The potential well minimum was found to be 5.36 eV at 2.36 Å. That is in a good agreement with a calculated value of 5.33 eV at 2.34 Å and an experimental value of 5.6 eV at 2.36 Å reported by Ref. 3. These values can also be found in Table 2. The potential curve from this simulation and Ref. 3 are plotted in Fig. 4.

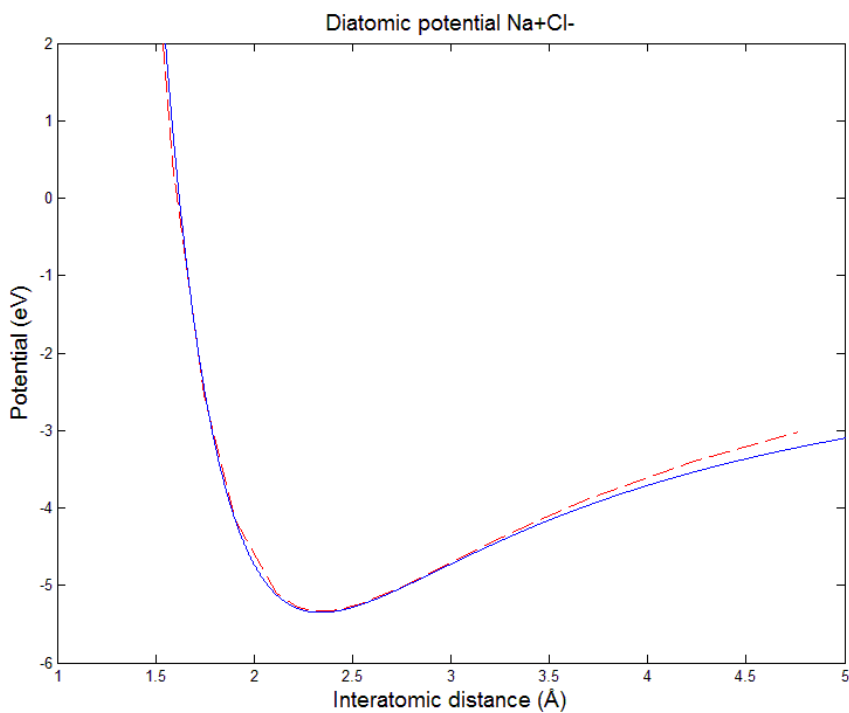


Fig. 4 --- calculated potential from Ref. 9; — current simulation.

4.2 Cohesive energy of NaCl

LAMMPS input: CohesiveNaCl.txt (Appendix B)
Matlab script: n/a
Timestep: n/a | Runtime: n/a

A NaCl crystal consisting of 8 atoms (1 unit cell) is relaxed to find the minimum energy and equilibrium separations. The maximal fractional change in volume of the simulation box that occurs in each iteration of the minimizer was set to 0.001. The simulation was done at 0 K and 0 bar.

Results and comparison with literature

The simulation provides the following output:

```
Total energy (eV) = -31.785770268261629;  
Lattice constant (Angstroms) = 5.574468253006879;  
Cohesive energy (eV) = -3.9732212835327037;
```

The cohesive energy shown here is simply the total energy divided by the number of atoms. We can therefore assume an average cohesive energy per ion *pair* of -7.95 eV. The lattice constant is determined to be 5.57 Å. In Ref. 9 we found a calculated potential value of -7.92 eV and an experimental value of 8.0 eV. The authors also report a calculated lattice constant of 5.72 Å and an experimental value of 5.64 Å. Values from this and the previous simulation are presented in Table 2 for comparison. Increasing the size of the NaCl crystal from 8 to 1000 atoms hardly increases accuracy; the cohesive energy was found to increase by less than 1E-5 eV.

	Potential well minimum (eV)			Equilibrium separation distance (Å)		
	This work	Calculated ^[9]	Experimental ^[9]	This work	Calculated ^[9]	Experimental ^[9]
Na+Cl-, diatomic	-5.36	-5.33	-5.6	2.36	2.34	2.36
Na+Cl-, crystal	-7.95	-7.92	-8.0	2.79	2.86	2.82

Table 2. NaCl properties found in this simulation compared to experimental and calculated data from Ref. 9.

4.3 Stress-strain curve of a cubic NaCl crystal

LAMMPS input: StressStrainNaCl.txt (Appendix C)
Matlab script: StressStrainplot.m (Appendix C)
Timestep: 1 fs | Runtime: 40 ps

A stress-strain simulation was done with the purpose of checking if the Young's Modulus of the simulated crystal corresponds to the literature value.

This simulation has two phases. The first phase is an equilibration phase of 20ps with a 1fs timestep. The 8000 ion sized crystal is allowed to expand until a stable state is reached in an isothermal–isobaric system (NPT ensemble) at a temperature of 300 K and pressure of 0 bar. The length of the simulation cell along the x-axis is determined at the end of this phase and stored as a variable for use in the deformation phase. This initial length will be used to calculate the strain.

In the second phase deformation takes place for 20 ps. Uni-axial tension is applied by deforming the lattice in the x-direction. The strain rate is 0.1 per picosecond (1000 time steps). The boundaries parallel to the direction of tension are controlled using the NPT equations of motion to keep them at zero pressure. Besides at 300 K, the simulation has also been run for 100 K and 500 K.

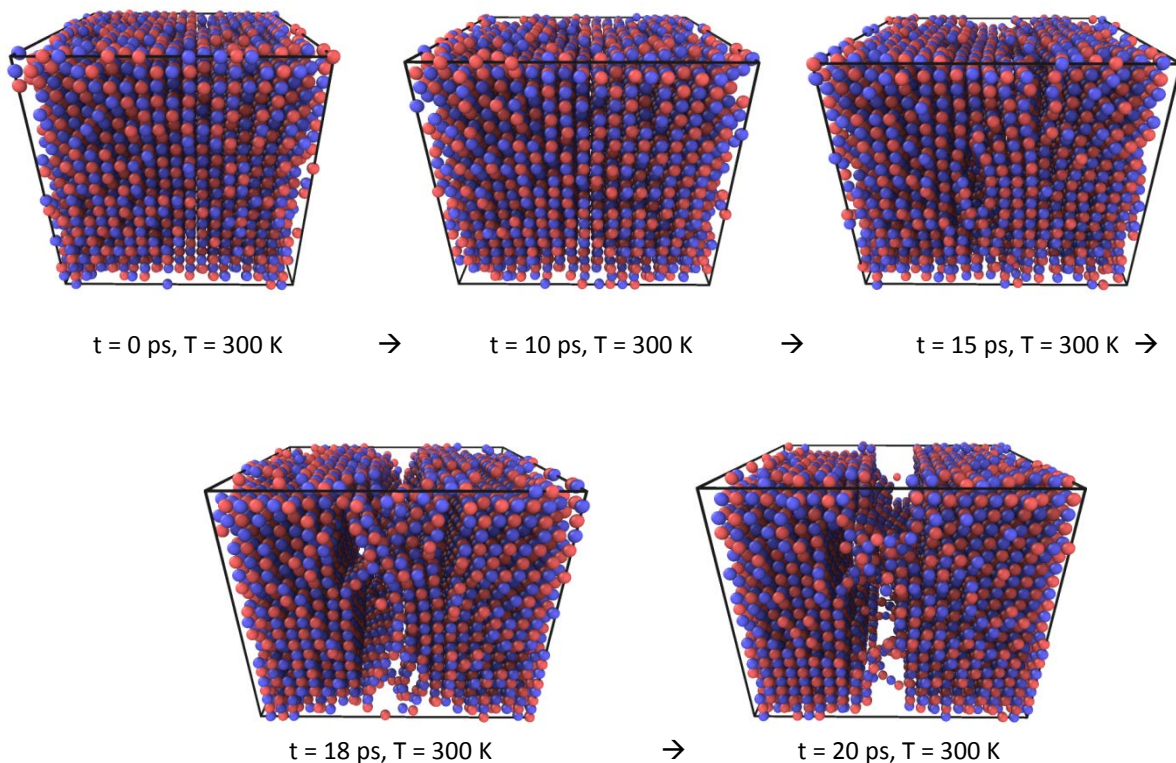


Fig. 5 Visualization of the deformation phase of NaCl under strain. Strain rate is 0.1 1/ps in the x-direction. The crystal becomes elongated and subsequently tears apart. Red spheres represent Na⁺, blue Cl⁻. Sphere radii are set to be equal for clearer images, without affecting the simulation.

Note that the Ewald summation value is very important in this simulation. I would advise at least $1.0E-3$ for reasonable results, and if time permits $1.0E-6$ for more accurate results.

Results and comparison with literature

Using a Matlab script to plot the stress as a function of the strain at different temperatures gives the stress-strain curves plotted in Fig. 6. The stress-strain curve behaves as expected for an ionic compound: the stress rises quickly in a relatively small elastic deformation region, indicating a brittle substance.

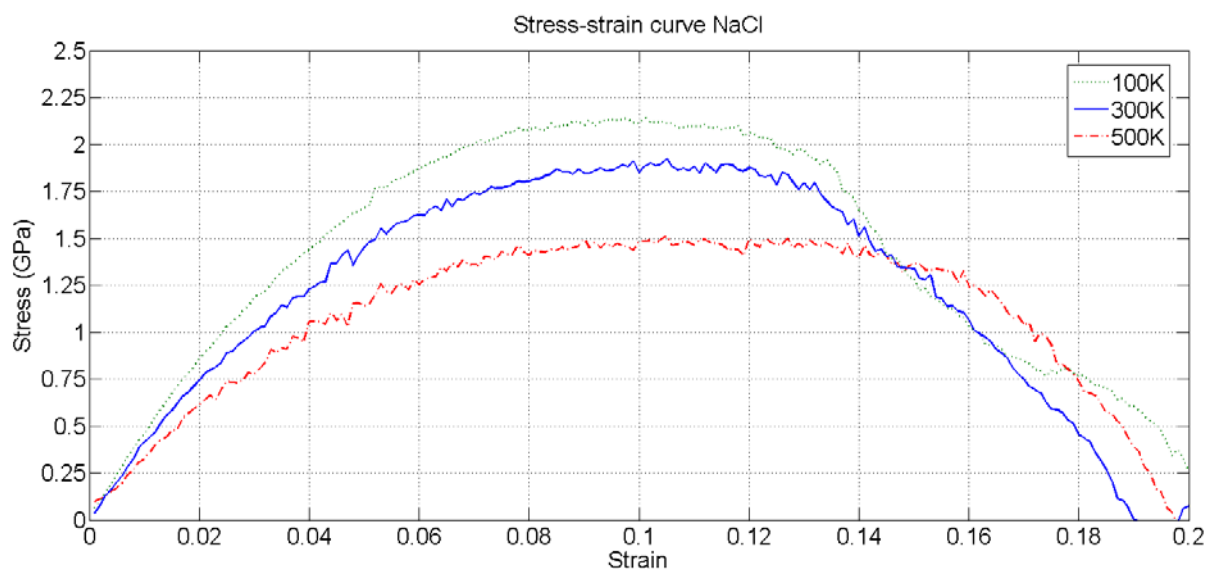


Fig. 6 Stress-strain curve for NaCl. 100 K; - 300 K; - - - 500 K.

The Young's Modulus at 300 K is found to be 39.56 GPa by taking the slope of a linear fit in the linear region (approximately 0 – 0.02 strain). The value found corresponds quite well with a literature value of 39.98 GPa^[10]. The curves in Fig. 6 are not entirely smooth due to temperature and/or pressure oscillations, which is a byproduct of using the Nosé–Hoover thermostat and barostat.

4.4 Li⁺ bombardment on NaCl

LAMMPS input: LiIonGun.txt (APPENDIX D)
Matlab script: n/a
Timestep: 0.05 fs | Runtime: 1 ps

This was the first simulation created and uses a different potential from the one used in all other simulations in this work (Buckingham). It is therefore not directly related to the main simulation (Chapter 5), but can still be used independently for other purposes. This simulation uses the Embedded Ion Method (EIM) developed by Zhou et. al ^[11] to simulate an ion gun firing Li⁺ ions at a NaCl crystal. We are interested in the sputter yield. The simulation was first run 20 times for 100 eV, resetting each time after impact and the sputter yield was counted. The same was then done for 200 eV.

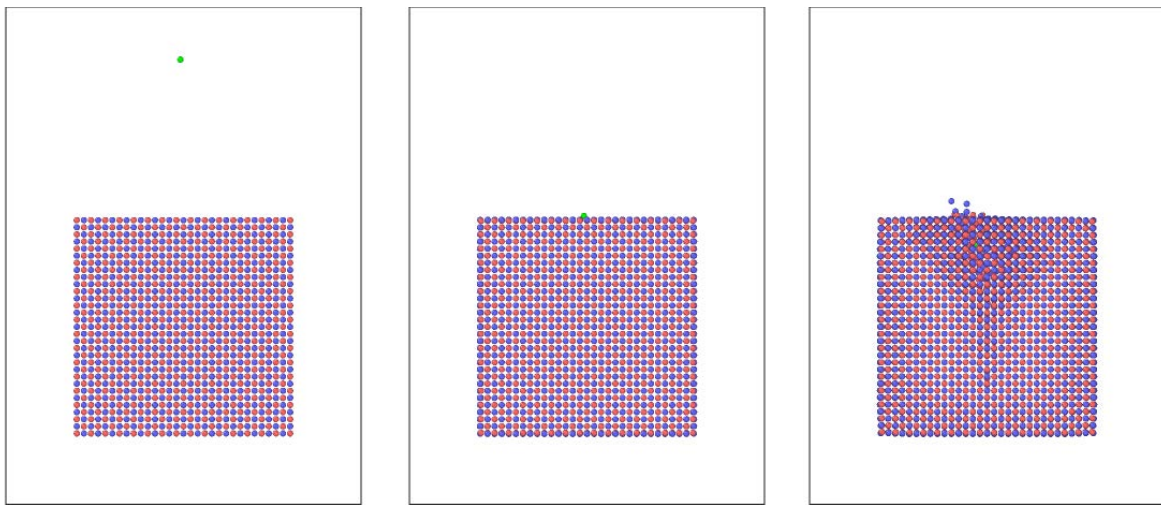


Fig. 7. Impact of lithium ion (green) on NaCl resulting in sputtering.

Results and comparison with SRIM 2013

The sputter yield is given in Table 3. The results are not very accurate, due to the limited number of simulation runs. It does however give a good indication that the results might agree with SRIM if more simulation runs would be performed.

	100 eV		200 eV	
	This work	SRIM	This work	SRIM
Na ⁺	0.25	0.2517	0.4	0.3775
Cl ⁻	0.2	0.1806	0.2	0.2766

Table 3. Sputter yield in number of ions per simulation run at 100 eV and 200 eV after Li⁺ ion impact

5. Main simulation:

H-center clustering and migration in NaCl

LAMMPS input: ClClustering.txt (Appendix E)
Matlab script: Clustering.m (Appendix E)

5.1 Simulation method

A simulation box of $15 \times 15 \times 15 a$ ($a = 5.57 \text{ \AA}$) was filled with Na^+ and Cl^- ions in the typical interleaved FCC sodium chloride lattice structure. The simulation was carried out in the NPT ensemble using an anisotropic pressure coupling. Neutral Cl atoms were randomly placed in a spherical region with radius of $4.5 a$ and the origin at $[7.5, 7.5, 7.5] a$. Two simulations were run: with low and high initial densities of the Cl atoms. In the case of the “low” density 40 atoms were placed in the spherical region, and for “high” density 120 atoms in the same region, giving $0.00013 \text{ atom/\AA}^3$ and $0.00039 \text{ atom/\AA}^3$ respectively. The total runtime in both cases was 200ps with a 5 fs time step. An equilibrium phase is not needed as long as the random seed is chosen in a way that the initial starting positions of the Cl atoms are not very close to an ion. If Cl is deposited close to an ion the resulting repulsion will impart a significant amount of kinetic energy to the Cl atom and associated ion, resulting in a collision cascade and sharp temperature rise for the entire system, which is obviously unwanted. The temperature is kept at 300 K using the Nosé–Hoover thermostat.

Clustering is quantified using the following approach. Each Cl atom has a unique atom-ID. When a Cl atom is within the cutoff range of another Cl atom, they will be assigned a cluster-ID which will be equal to the lowest numbered atom-ID in the cluster. The cutoff range is set to 4 \AA . The cluster migration is followed by calculating the center of mass for each cluster at different times during the simulation.

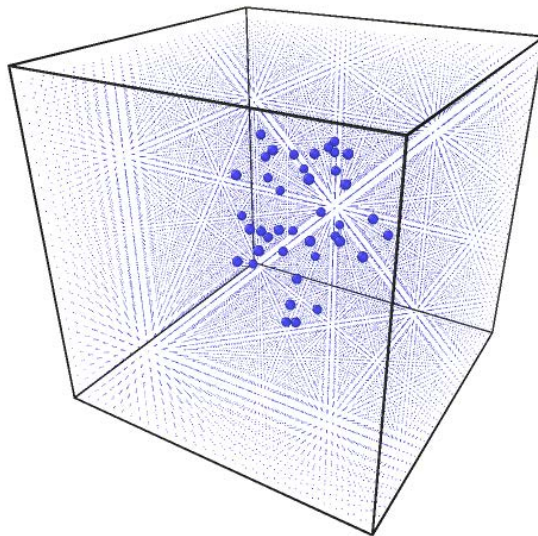


Fig. 8. Low density scenario at 0 ps. Na^+ and Cl^- radii are reduced to 0.1 and $r(\text{Cl}) = 1.25$

5.2 Potentials

For this simulation three new potentials are needed to describe the interatomic interactions for Na⁺-Cl, Cl-Cl and Cl-Cl. We use a Buckingham potential in combination with a Coulomb potential.

Cl-Cl interaction

A Moliere potential and associated parameters were found for the interatomic potential of two neutral Cl atoms in Ref. 12.

$$U(r) = a \cdot (b \cdot r^{-p} - r^{-q}) \cdot \exp\left(\frac{c}{r - r_c}\right), \quad r < r_c \quad (2)$$

The parameters used with this potential cannot be used with a Buckingham potential as is. The Moliere potential was therefore plotted and the Buckingham potential was solved at 3 points for 3 unknown variables (A, B, C). The key points were chosen to be the deepest point of the potential well, a point near crossing the x-axis and a point near the x-axis at a bigger separation distance (3.5 Å – 4 Å).

The *best fitting* coefficients are $A = -367.4$, $B = 0.5005$, $C = -275.05$. The author is aware that negative values for the repulsive part and Van der Waals coefficient do not make physical sense, they do however provide the best fit possible, so are used regardless. Another Cl-Cl interaction potential (and parameters) was found in Morse form in Ref. 13.

$$U(r) = D \left\{ 1 - \exp\left[\frac{(r - r_e)}{B}\right] \right\} \quad (3)$$

The Moliere potential, Morse potential and the fit used for the simulation are plotted in Fig. 9.

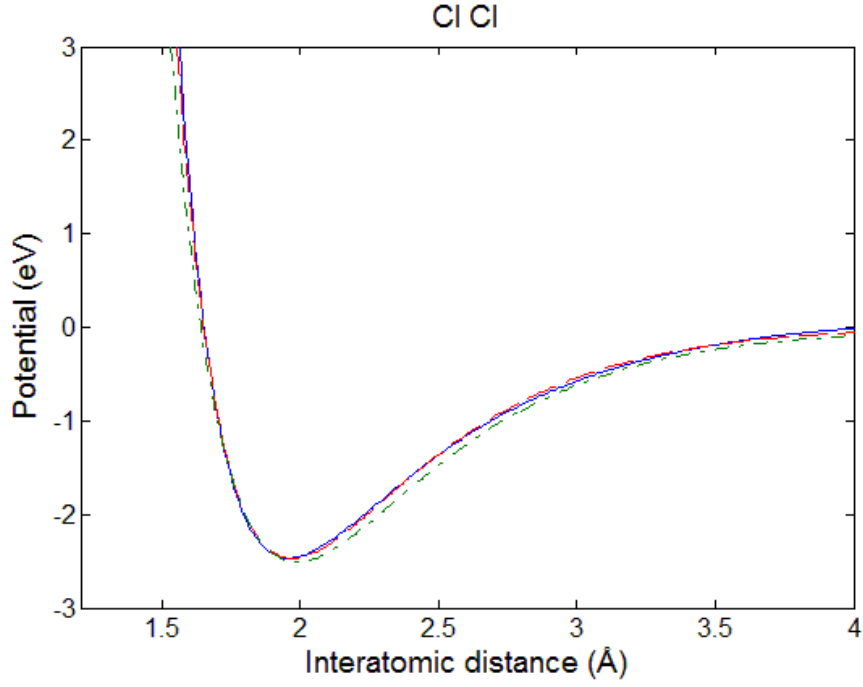


Fig. 9. — this work. - - - Moliere potential from Ref. 12. - · - potential from Ref. 13

Cl⁻-Cl and Na⁺-Cl interaction

The parameters for Cl⁻-Cl and Na⁺-Cl interaction are presented in Ref. 13 in Born-Mayer form.

$$U(r) = A \cdot \exp\left(-\frac{r}{B}\right) \quad (4)$$

Note that this entire potential is the same as the second term in equation (1). We can therefore use Born-Mayer potentials in our simulation without adding or altering any code and simply set $C = 0$. All potentials used in this simulation are those from Table 1 and Table 4.

Interactions	A (eV)	B (Å)	C (eV Å ⁶)
Na ⁺ -Cl	1012.6	0.2903	0.00
Cl ⁻ -Cl	1205.0	0.3358	0.00
Cl-Cl	-367.4	.5005	-275.1
$Z_{\text{Na}^+} = +0.988$		$Z_{\text{Cl}^-} = -0.988$	

Table 4: Potentials from ref. 13 and fitted potential for Cl-Cl

5.3 Results

Clustering of Cl atoms is indeed observed in the simulation as expected. In Fig. 10b and Fig. 11b two graphs are presented showing the amount and size of clusters at different times: one at the beginning of the simulation, one at 20 ps, 100 ps and 200 ps. Each color bar represents a certain time. There is no clear movement pattern observed by visually inspecting the simulation. The center of mass coordinates for each cluster are not presented here, since they give no useful additional information. For future work, a MATLAB script to calculate the coordinates for each cluster can be found in appendix E.

Low density:

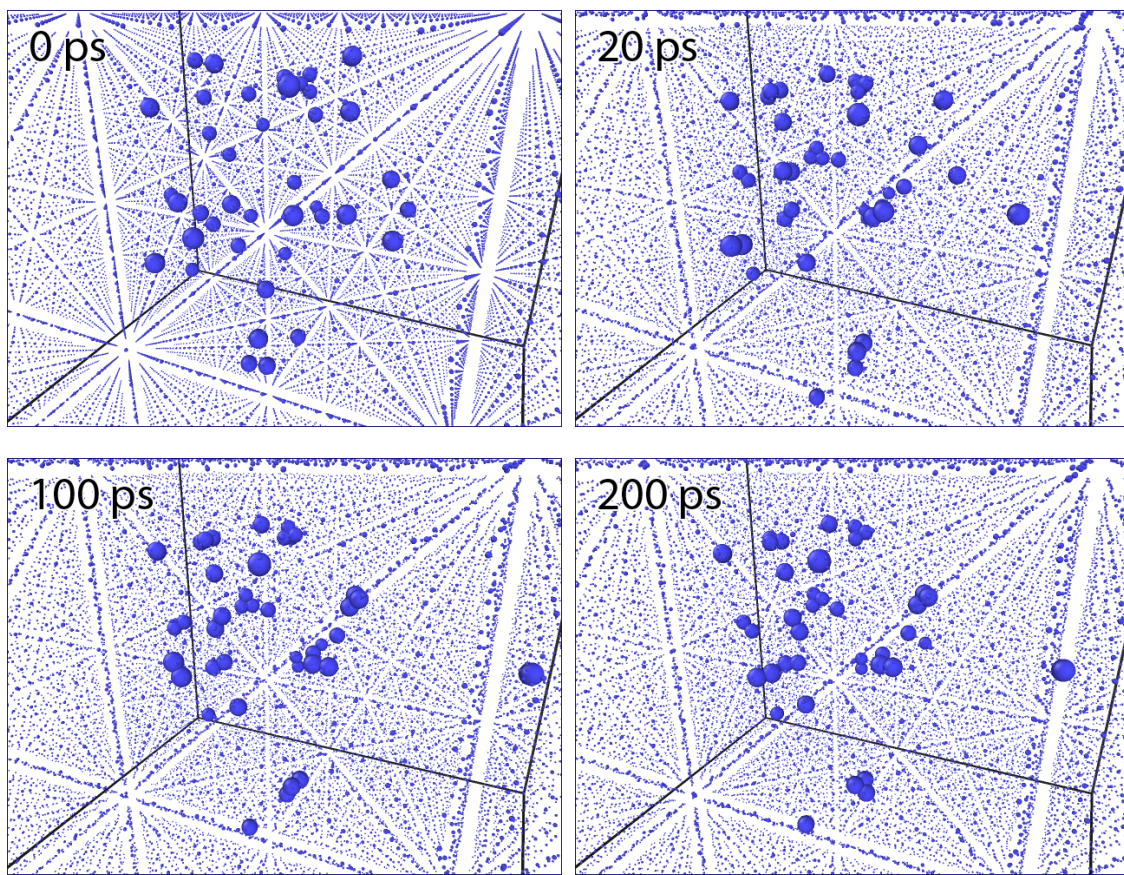


Fig. 10a. Visualization of clustering over 200ps for the low-density scenario.

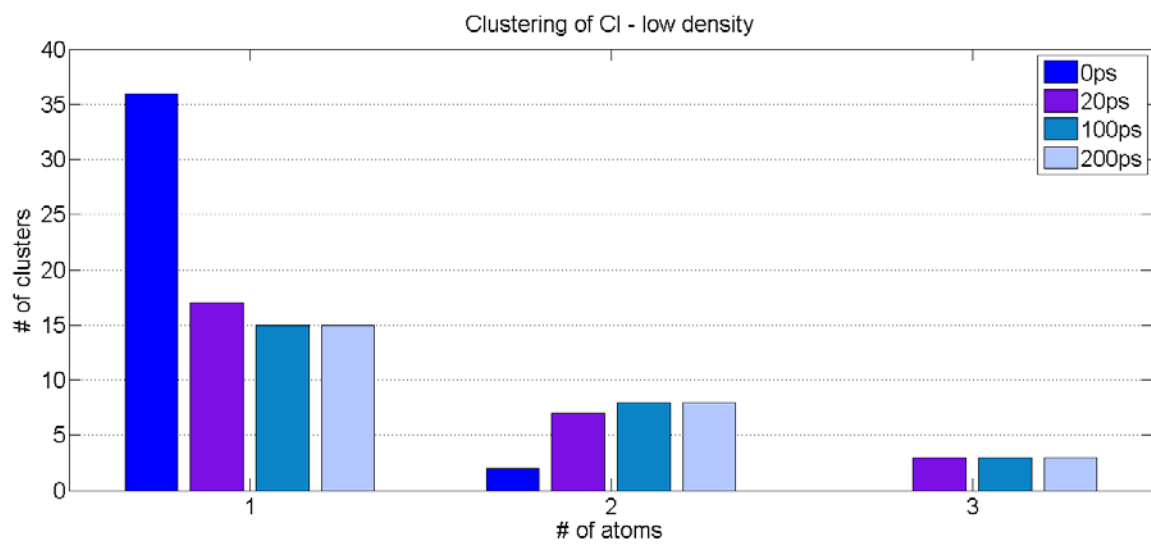


Fig. 10b. Number of clusters containing a certain number of atoms for the low-density scenario.

High density:

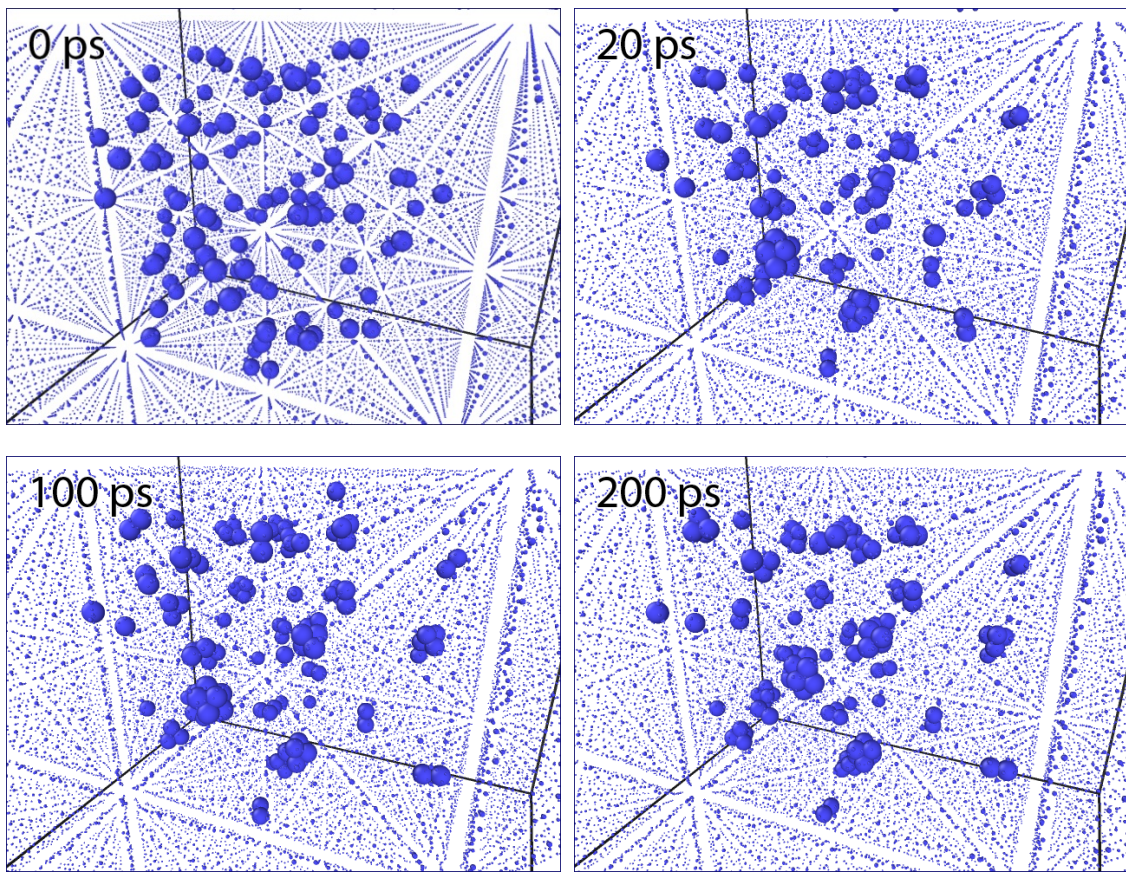


Fig. 11a. Visualization of clustering over 200 ps for the high-density scenario.

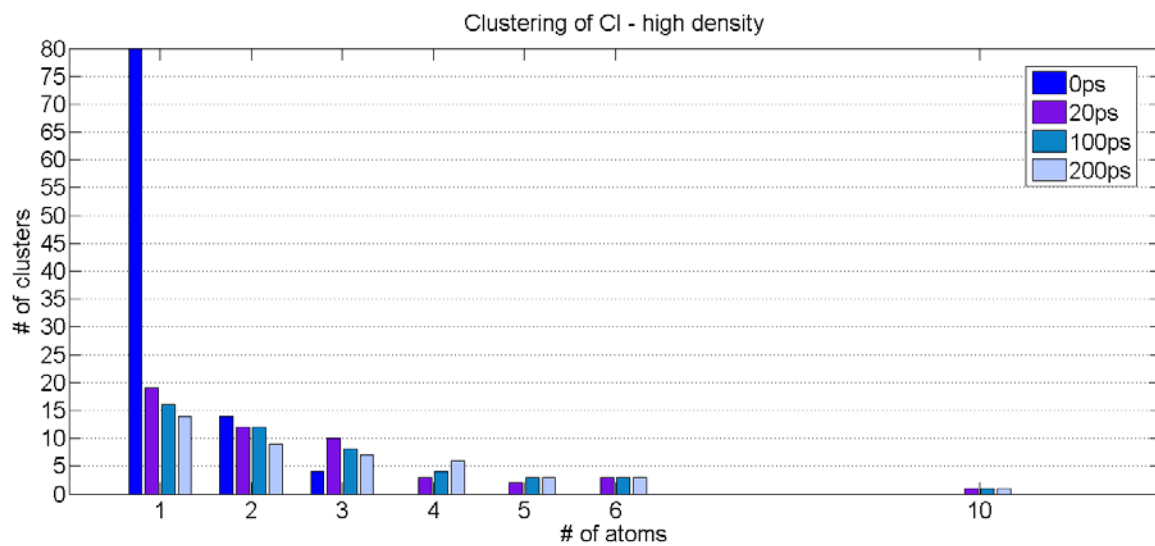


Fig. 11b. Number of clusters containing a certain number of atoms for the high-density scenario.

6. Conclusion & Discussion

From the preliminary simulations described in this paper we can conclude that the simulated NaCl crystal behaves as expected. The diatomic energy, cohesive energy and Young's modulus all agree with experimental and calculated values from literature.

The main simulation shows clustering does occur as expected from the ionoluminescence results on the He⁺ ion bombardment of NaCl crystals in the HIM. Most clustering occurs during the first 20 ps. The clusters move around slightly, but there is no clear migration observed in any direction, nor is there any more clustering. Lone neutral Cl atoms not belonging to a cluster are more mobile, but also do not show a preferential diffusion direction. This may have to do with the runtime of the simulation. Experimentally the migration is observed on a timescale of microseconds, so there is a difference in order of magnitude of 10⁶. It may simply be the case that the timeframe of the simulation was too short to detect any movement patterns. It may also be the case that with different initial Cl densities, the migration behavior may be different.

For future work it would be suggested to increase the runtime of the simulation to μ s scale. To make this feasible, the timestep should be increased considerably and the Ewald summation number should be increased (to 1.0E-1 for example). Other ways of optimization should also be looked into, since the 200 ps simulations described in this paper took about 12 hours of computing time on a quad-core computer. Making no alterations to the input script and simply running a μ s simulation is unrealistic, since it would literally take years of computing time. Another suggestion is to observe the system development at different initial Cl densities; this would be possible without any major alterations to the input script.

This simulation does not account for the presence of α -centers and this may have an impact on H-center migration. α -centers are easily added to the simulation by deleting a certain amount of atoms of a certain type with the LAMMPS command "delete_atoms". A simulation that includes α -centers was not run due to time constraints. This simulation also does not describe the presence or creation of F-centers, or its recombination with H-centers. It would be interesting to see the effects on the H-centers' diffusion if the simulation was expanded to include this. Since the electron is not centered in the vacancy it occupies, perhaps this may have an effect on the lattice ions and thereby the Cl migration.

Including F-centers in this simulation will provide several challenges however. Electrons may move at very high velocity even in such a relatively low energy system because of their limited mass. This creates challenges if they are simply modeled as point-particles. A very small timestep - much smaller than 1fs - should therefore be used in order to accurately calculate their interactions in the system. As mentioned before however, this would

make a μs simulation unfeasible due to the immense computing time required. Perhaps a simulation using a different approach would be more suitable for this, such as using the electron Force Field (eFF) potential currently under development at Caltech^[14]. eFF is an atom-style supported by LAMMPS which can be seen to be an approximation to QM wave packet dynamics and Fermionic molecular dynamics. The exact workings of this atom-style are beyond the scope of this work however. Other atom-styles may also be worth investigating in an effort to include F-centers.

7. References

- [1] Ziegler, J. F., SRIM - The Stopping and Range of Ions in Matter, <http://www.srim.org/> (17-7-2013)
- [2] Schulman J.H., 1963, *Color centers in solids (International series on monographs in solid state physics)*, Pergamon Press, 368 p.
- [3] Bergsma, J., and Heuboer, R. J., 1983, Radiation dose deposition and energy accumulation in a rock salt waste repository, *ECN*, vol. 144, p. 1-7
- [4] Sonder, E., 1973, Trapping of mobile interstitials during irradiation of alkali halides, *Journal de Physique*, vol. 34, p. 32-37
- [5] Zhigilei, L., Introduction to Atomistic Simulations, <http://people.virginia.edu/~lz2n/mse627/notes/Potentials.pdf> (15-7-2013)
- [6] Sakib, A. R. N., and Adnan, A., 2012, On the size-dependent critical stress intensity factor of confined brittle nanofilms, *Engineering Fracture Mechanics*, vol. 86, p. 13-22, ISSN 0013-7944
- [7] Stukowski, A., OVITO - The Open Visualization Tool, <http://www.ovito.org/> (17-7-2013)
- [8] Plimpton S., LAMMPS Documentation, <http://lammps.sandia.gov/doc/Manual.html> (15-7-2013)
- [9] Gordon, R. G., and Kim, Y. S., 1972, Theory for the Forces between Closed-Shell Atoms and Molecules, *Journal of Chemical Physics*, vol. 56, 3122
- [10] Crystran Ltd. Poole, Sodium Chloride (NaCl) Materials Data, <http://www.crystran.co.uk/userfiles/files/sodium-chloride-nacl-data-sheet.pdf> (15-7-2013)
- [11] Zhou, X. W., and Doty, F. P., 2008, Embedded-ion method: An analytical energy-conserving charge-transfer interatomic potential and its application to the La-Br system, *Physical Review B*, vol. 78, 224307
- [12] Feil, H., Dieleman, J., and Garrison, B. J., 1993, Chemical sputtering of Si related to roughness formation of a Cl-passivated Si surface, *Journal of Applied Physics*, vol. 74, 1303
- [13] Catlow, C. R. A., Diller, K. M., Hobbs, L. W., 1980 Irradiation-induced defects in alkali halide crystals, *Philosophical Magazine A*, vol. 42, issue 2, p. 123-150, DOI: 10.1080/01418618009365806.
- [14] Jaramillo-Botero, Su, Qi, Goddard, 2011, Large-scale, Long-term Non-adiabatic Electron Molecular Dynamics for Describing Material Properties and Phenomena in Extreme Environments, *Journal of Computational Chemistry*, vol. 32, 497-512

Acknowledgements

The author wishes to thank several people. I would like to first and foremost thank my supervisors Vasilisa Veligura and Gregor Hlawacek for their incredibly helpful advice, assistance and guidance during this assignment. The time and energy spent is very greatly appreciated. I would also like to thank Herbert Wormeester very much for his time and effort in helping me during the initial experimental assignment. Thanks also to Stefan Kooij for accepting my request to do my Bsc. Assignment at the PIN-group and providing the computer on which the simulations in this paper were created and run, it's very much appreciated. Also many thanks to Wouter den Otter for agreeing to be part of my bachelor assignment committee and for being available for MD specific questions. Lastly I would like to show my appreciation to the other members of the PIN-group at Utwente for their hospitality during my two and a half month stay.

Appendix A: Diatomic energy and separation of Na⁺ Cl⁻ scripts

Diatomicpotential.txt

```
# Diatomic potential in NaCl
# Karun Datadien
# Latest update: 18-6-2013

# ----- INITIALIZATION -----
units          metal
dimension      3
boundary       p p p
atom_style     hybrid atomic charge bond

# ----- ATOM DEFINITION -----

lattice        fcc 5.64 origin 0 0 0
region         sample block -10 30 1 3 1 3
create_box    2 sample

lattice        fcc 5.64 origin 0 0 0
create_atoms   1 single 9 2 2
create_atoms   2 single 9.1 2 2

# Set masses
mass 1 22.98976928          #atomic weight sodium
mass 2 35.453              #atomic weight of chlorine

# Set charges
set type 1 charge +0.988
set type 2 charge -0.988

# Misc.
neighbor 0.3 bin
neigh_modify delay 0

# ----- FORCE FIELDS -----
pair_style     buck/coul/long 10.0 8.0
pair_coeff     1 1 7895.4 0.1709 29.06          #Na+ Na+
pair_coeff     2 2 1227.2 0.3214 29.06          #Cl- Cl-
pair_coeff     1 2 2314.7 0.2903 0.0           #Na+ Cl-
kspace_style   ewald 1.0E-5

#Temperature
group         samplegroup region sample
fix           int samplegroup nvt temp 1.0 1.0 5.0

# ----- COMPUTING -----
compute csym all centro/atom 6          # 6 nearest neighbours for two interpenetrating lattices
compute peratomPE all pe/atom
compute peratomKE all ke/atom
```

```

compute tempcompute samplegroup temp
compute KineticEnergy samplegroup ke/atom

# ----- DATA OUTPUT -----

dump          newdump all cfg 10 dump.ABuckCpotential*.cfg id type xs ys zs c_csym
c_peratomPE c_peratomKE fx fy fz mass vz c_KineticEnergy

dump_modify          newdump pad 2 element Na Cl

# Display thermo
timestep             .0000001          #.001 = femtosecond
thermo_style         custom step pe ke temp press c_tempcompute
thermo              100

run                 200000

# ----SIMULATION DONE-----
print "Simulation complete"

```

DPotentialgraph.m

```

load BuckCafstanden2;          % load data
set(0, 'DefaultAxesColorOrder', [0 0 0], ...
      'DefaultAxesLineStyleOrder', '-|--|-.|:');

% experimental data:
y1 = 27.2116*[0.1277D04 0.8631D02 0.1535D02 0.4024D01 0.1349D01
0.4060D00 0.2085D00 0.1670D-1 -0.9339D-1 -0.1530D00 -0.18748D00 -0.19396D00
-0.19535D00 -0.19587D00 -0.19588D00 -0.19575D00 -0.19510D00 -0.19255D00 -
0.18886D00 -0.18449D00 -0.18215D00 -0.17974D00 -0.17482D00 -0.16490D00 -
0.15315D00 -0.14260D00 -0.12499D00 -0.11112D00];
x1= 0.529177249*[0.1 0.5 1.0 1.5 2.0 2.5 2.7 3.0 3.3 3.6 4.0 4.2 4.3
4.4 4.45 4.5 4.6 4.8 5.0 5.2 5.3 5.4 5.6 6.0 6.5 7.0 8.0 9.0] ;

for i = 1:2:40002

x2(i,:) = BuckCafstanden2(i+1,1) - BuckCafstanden2(i,1);
y2(i,:) = BuckCafstanden2(i+1,2) + BuckCafstanden2(i,2);

end

x2((2:2:end), :) = [];          % delete every other row (=0)
y2((2:2:end), :) = [];

plot(x1, y1, x2, y2);
xlabel('Interatomic distance (Å)'); % add axis labels and plot title
ylabel('Potential (eV)');
title('Diatomic potential Na+Cl-');

```

Appendix B: Cohesive energy of NaCl script

CohesiveNaCl.txt

```
# Find minimum energy configuration
# ----- Initialize Simulation -----
clear
units          metal
dimension      3
boundary       p p p
atom_style     hybrid atomic charge

# ----- Create Atoms -----

# Na atoms
lattice        fcc 5.7 origin 0 0 0          # Lattice for Na system
region         sample block 0 1.0 0 1.0 0 1.0 # x-begin x-eind y-begin y-eind z-begin z-eind

create_box     2 sample

#Na+
lattice        fcc 5.64 origin 0 0 0
create_atoms   1 region sample

#Cl-
lattice        fcc 5.64 origin .5 0.5 0.5
create_atoms   2 region sample

# Set masses
mass 1 22.98976928      #atomic weight sodium
mass 2 35.453           #atomic weight of chlorine

# Set charges
set type 1 charge +0.988
set type 2 charge -0.988

# ----- Define Interatomic Potential -----
pair_style     buck/coul/long 10.0 8.0
pair_coeff     1 1 7895.4 0.1709 29.06      #Na+ Na+
pair_coeff     2 2 1227.2 0.3214 29.06      #Cl- Cl-
pair_coeff     1 2 2314.7 0.2903 0.00        #Na+ Cl-
kspace_style   ewald 1.0E-6

neighbor 2.0 bin
neigh_modify delay 10 check yes

group         samplegroup region sample
# ----- Define Settings -----
compute eng all pe/atom
compute eatoms all reduce sum c_eng
compute tempcompute samplegroup temp

# ----- Run Minimization -----
reset_timestep 0
```



```
fix                          1 all box/relax iso 0.0 vmax 0.001
thermo                        10
thermo_style                  custom step pe lx ly lz press pxx pyy pzz c_eatoms c_tempcompute
min_style                     cg
minimize                      1e-25 1e-25 5000 10000
```

```
variable natoms equal "count(all)"
variable teng equal "c_eatoms"
variable length equal "lx"
variable ecoh equal "v_teng/v_natoms"
```

```
print "Total energy (eV) = ${teng};"
print "Number of atoms = ${natoms};"
print "Lattice constant (Angstroms) = ${length};"
print "Cohesive energy (eV) = ${ecoh};"
```

Appendix C: Stress-strain curve of a cubic NaCl crystal scripts

StressStrainNaCl.txt

```
# ----- INITIALIZATION -----
units          metal
dimension3
boundary p      p      p
atom_style      hybrid atomic charge

# ----- ATOM DEFINITION -----

# Na atoms
lattice         fcc 5.57 origin 0 0 0          # Lattice for Na system
region         sample block 0 10.0 0 10.0 0 10.0 # x-begin x-eind y-begin y-eind z-begin z-eind

create_box      2 sample

#Na+
lattice         fcc 5.57 origin 0 0 0
create_atoms    1 region sample

#Cl-
lattice         fcc 5.57 origin .5 0.5 0.5
create_atoms    2 region sample

# Set masses
mass 1 22.98976928 #atomic weight sodium
mass 2 35.453      #atomic weight of chlorine

# Set charges
set type 1 charge +0.988
set type 2 charge -0.988

# ----- FORCE FIELDS -----
pair_style      buck/coul/long 10.0 8.0
pair_coeff      1 1 7895.4 0.1709 29.06      #Na+ Na+
pair_coeff      2 2 1227.2 0.3214 29.06      #Cl- Cl-
pair_coeff      1 2 2314.7 0.2903 0.00       #Na+ Cl-
kspace_style    ewald 1.0E-4

neighbor 2.0 bin
neigh_modify delay 10 check yes

# ----- SETTINGS -----
compute csym all centro/atom 6
compute peratom all pe/atom

#####
# EQUILIBRATION
reset_timestep  0
timestep 0.001
velocity all create 500 12345 mom yes rot no
```

```

fix 1 all npt temp 500 500 1 iso 0 0 1 drag 1

# Set thermo output
thermo 1000
thermo_style custom step lx ly lz press pxx pyy pzz pe temp

# Run for at least 10 picosecond (assuming 1 fs timestep)
run 20000
unfix 1

# Store final cell length for strain calculations
variable tmp equal "lx"
variable L0 equal ${tmp}
print "Initial Length, L0: ${L0}"

#####
# DEFORMATION
reset_timestep 0

fix 1 all npt temp 500 500 1 y 0 0 1 z 0 0 1 drag 1
variable srate equal 1.0e10
variable srate1 equal "v_srate / 1.0e12"
fix 2 all deform 1 x erate ${srate1} units box remap x

# Output strain and stress info to file
# for units metal, pressure is in [bars] = 100 [kPa] = 1/10000 [GPa]
# p2, p3, p4 are in GPa
variable strain equal "(lx - v_L0)/v_L0"
variable p1 equal "v_strain"
variable p2 equal "-pxx/10000"
variable p3 equal "-pyy/10000"
variable p4 equal "-pzz/10000"
fix def1 all print 100 "${p1} ${p2} ${p3} ${p4}" file NaClStrain500k.def1.txt screen no

# Use cfg for AtomEye
dump 1 all cfg 250 dump.tensile500k_*.cfg id type xs ys zs c_csym c_peratom fx fy fz
dump_modify 1 pad 2 element Na Cl

# Display thermo
thermo 1000
thermo_style custom step v_strain temp v_p2 v_p3 v_p4 ke pe press

run 25000

#####
# SIMULATION DONE
print " Simulation complete "

```

StressStrainplot.m

```
d = dir('C:\LAMMPS\NaClStrain500k.def1.txt'); % load data

for i = 1:length(d)

    fname = d(i).name
    A = importdata(fname)
    strain = A.data(:,1)
    stress = A.data(:,2)

    plot(strain, stress)
    xlabel('Strain'); % add axis labels and plot title
    ylabel('Stress');
    title('Stress-strain curve NaCl');

end
```

Appendix D: Li⁺ bombardment on NaCl scripts

LiIonGun.txt

```
# Karun Datadien
# Latest update: 18-6-2013

# ----- INITIALIZATION -----
units          metal
dimension3
boundary f f f
atom_style      hybrid atomic charge bond

processors * * *                               # Use multiple cores

# ----- ATOM DEFINITION -----
# Na atoms region
lattice         fcc 5.64 origin 0 0 0          # Na lattice
region          sample block 0 15 0 15 0 15    # x-begin x-eind y-begin y-eind z-begin z-eind

# Li atoms region
region          beam block 6 9 6 9 16 17      # uses the same lattice as above

# Simulation box size
region          boundary block -5 20 -5 20 -5 25

#combine regions
region          whole union 3 sample beam boundary

#create viewing box
create_box      3 boundary

lattice         fcc 5.64 origin 0 0 0
create_atoms    1 region sample
lattice         fcc 5.64 origin .5 0.5 0.5
create_atoms    2 region sample

create_atoms    3 random 1 111422 beam        #region beam

# Set wall
fix wall all wall/region whole lj93 5 5 5

# Misc.
neighbor 0.3 bin
neigh_modify delay 0

# ----- FORCE FIELDS -----
pair_style eim

pair_coeff * * Na Cl Liffield.eim Na Cl Li

#Temperature
```

```

group          samplegroup region sample
fix           temp3 samplegroup nvt temp 1.0 1.0 5.0

#Beam velocity
group         beamgroup region beam
velocity      beamgroup set 0 0 -95          # !!! In lattice constants: a/ps
fix           beam3 beamgroup nve

# ----- COMPUTING -----
# Compute the data we want in the output files

compute csym all centro/atom 6              # 6 nearest neighbours for two interpenetrating lattices
compute peratom all pe/atom                 # Define a computation that computes the per-atom
                                             # potential energy for each atom in a group
compute tempcompute samplegroup temp       # Define a computation that calculates the temperature of
                                             # a group of atoms.
compute KineticEnergy all ke/atom

# ----- DATA OUTPUT -----
# Create dumpfiles and other output. Specify timescale and total runtime.

dump          mydump all cfg 500 dump.IonGunEIM_*.cfg id type xs ys zs c_peratom fx fy
fz mass vz c_KineticEnergy #c_BondEnergy

dump_modify   mydump pad 3 element Na Cl Li

# Display thermo
timestep      .00005                        #.001 = femtosecond
thermo_style  custom step pe pxx pyy pzz temp press c_tempcompute
thermo        100

run           20000                          #run x timesteps

# ----SIMULATION DONE-----
print "Simulation complete"

```

Appendix E: Main simulation + clustering scripts

ClClustering.txt

```
# H-center clustering in NaCl
# Karun Datadien | last update: 11-7-2013

# ----- INITIALIZATION -----
units          metal
dimension3
boundary p p p
atom_style     hybrid atomic charge

# ----- ATOM DEFINITION -----
# Simulation box
lattice        fcc 5.57 origin 0 0 0
region        sample block 0 15.0 0 15.0 0 15.0
region        DefectRegion sphere 7.5 7.5 7.5 4.5

create_box     3 sample

#Na+
lattice        fcc 5.57 origin 0 0 0
create_atoms   1 region sample

#Cl-
lattice        fcc 5.57 origin .5 0.5 0.5
create_atoms   2 region sample

#Cl (neutral)
create_atoms   3 random 40 789465 DefectRegion
group         NeutralClgroup type 3

#Set masses
mass 1 22.98976928      #atomic weight sodium
mass 2 35.453          #atomic weight of chlorine
mass 3 35.453          #atomic weight of neutral chlorine

#Set charges
set type 1 charge +0.988
set type 2 charge -0.988
set type 3 charge 0.0

#Misc.
neighbor 0.3 bin
neigh_modify delay 0

# ----- FORCE FIELDS -----

pair_style     buck/coul/long 10.0 8.0
pair_coeff     1 1 7895.4 0.1709 29.06      #Na+ Na+
pair_coeff     2 2 1227.2 0.3214 29.06      #Cl- Cl-
pair_coeff     1 2 2314.7 0.2903 0.00       #Na+ Cl-
```

```

pair_coeff          3 3 -367.4337814 .5005465005 -275.0537438      #Cl Cl
pair_coeff          2 3 1205 .3358 0                               #Cl- Cl
pair_coeff          1 3 1012.6 .2903 0                             #Na+ Cl

kpace_style        ewald 1.0E-3

#Temperature
group              samplegroup region sample
velocity           samplegroup create 300.0 43454 dist gaussian mom yes
fix                int samplegroup npt temp 300.0 300.0 .01 aniso 1.0 1.0 0.1

# ----- SETTINGS -----
compute peratom all pe/atom
compute Clustercount NeutralClgroup cluster/atom 4
compute 1 all temp

#####

# Dump cfg
dump 1 all cfg 100 dump.NeutralClFINALlowdensity_*.cfg id type xs ys zs c_peratom fx fy fz
c_Clustercount dump_modify 1 pad 3 element Na Cl Cl_Neutral

# Display thermo
timestep           .005          #.001 = femtosecond
thermo_style       custom step pe pxx pyy pzz temp press
thermo             50

run                40000

# -----SIMULATION DONE-----
print "Simulation complete"

```


Clustering.m

```
% Clustering & C.O.M. calculations for OVITO export files
% Karun Datadien | last-update: 16-7-2013
%
% To use this program, only a few settings need to be made in the
% first few lines of code.
%
% 1. load the proper file. This is an exported 'xyz-file' from OVITO
% where all rows with atom types other than 3 - indicating Cl - are
% deleted. The file must be in the same directory as the matlab
% script.
% 2. change variables to the columns in which the
% appropriate values are stored.
%
% For example, the CLUSTERLODENSITY100ps file contains 5 columns:
% Atom-ID | X-pos | Y-pos | Z-pos | Cluster-ID
%
% So Xcol = 2, Ycol = 3, etc.

load CLUSTERLODENSITY100ps;           % Load data.
Xcol = 2;
Ycol = 3;
Zcol = 4;
Ccol = 5;

R = length(CLUSTERLODENSITY100ps(:,1));

for i = 1:1:R                          %40 = number of atoms / rows in loaded file
C(i,:) = CLUSTERLODENSITY100ps(i,Ccol); %get list of cluster-Id's
end

[h1 h2] = hist(C(:),unique(C));
h3 = unique(h1);
h3 = h3(end:-1:1);

%Gives the amount of atoms in each cluster:
for k=1:length(h3)
    disp(['There are ' num2str(h3(k)) ' atoms in these clusters: [ '
num2str(h2(h1==h3(k))) ' ]'])
end

y=h1;
[f1 f2] = hist(y(:),unique(y));
f3 = unique(f1);
f3 = f3(end:-1:1);

%plot
figure; bar(f2,f1,0.5,'r');
    xlabel('# of atoms');                % add axis labels and plot title
    ylabel('# of clusters');
    title('Clustering of Cl');
```

```

%The next part is where a list with center of mass coordinates is built

V=0;
Clength = 0; %number of atoms in the cluster

for V = 1:1:R
Clength = 0;

    for i = 1:1:R

        if(C(V,1) == C(i,1))
            X(i,:) = CLUSTERLODENSITY100ps(i,Xcol);
            Y(i,:) = CLUSTERLODENSITY100ps(i,Ycol);
            Z(i,:) = CLUSTERLODENSITY100ps(i,Zcol);
        else X(i,:) = 0; Y(i,:) = 0; Z(i,:) = 0;

        end

    end

%finds out number of atoms in the cluster
for i = 1:1:R

    if(X(i,:) > 0)
        Clength = Clength+1;
    else Clength = Clength;
    end

end

a=sum(X)/Clength; %COM X-direction of cluster
b=sum(Y)/Clength; %Y
c=sum(Z)/Clength; %Z

disp(['Center of mass (x,y,z) for cluster ' num2str(C(V,1)) ' is:
[ ' num2str([a b c]) ' ]'])

COM(V,:) = [a b c];

end

COM %output a list with the x,y,z C.O.M. coordinates of each cluster.

```