# Molecular Dynamic Simulations of Copper using Moldy

James Adler, Cornell University
Research Experience for Undergraduates 2003, National High Magnetic Field Laboratory
Advisors: Dr. Ke Han, NHMFL, Tallahassee, Florida
Mr. Jingping Chen, NHMFL, Florida State University

---

**Abstract**

This report documents a study of how the Moldy molecular dynamics program works using Argon as an example. The Moldy was also used to seek the possibility to simulate pure FCC copper. Crystal structure of FCC copper was created and simulations at temperatures of 1, 77, 300, and 900 Kelvin were run. Results showed that the Moldy program was consistent with experimental data.

---

## Background

Alder and Wainwright first introduced molecular Dynamics (MD) in the late 1950's to study the interaction of hard spheres. Many insights were made about the interactions of simple liquids. In 1964, a real potential was used to simulate liquid argon, and in 1974 the first realistic simulation was performed to study water. Since then molecular dynamics simulations have been used to study proteins, DNA complexes, and other biological research. Recently, mixed quantum mechanical and classical simulations are being used to study molecular dynamics including experimental procedures, such as X-ray crystallography and NMR (nuclear magnetic resonance) structure determination [1]. Our use for MD includes simulations of pure copper, which is used as conducting material in high-powered magnets. Further studies of copper with dislocations will be useful.

MD is a method for studying classical statistical mechanics of well-defined systems through numerical solutions of Newton's equations of motion. Ideally we would want to use quantum mechanics and solve the Schroedinger Equation (SE) to determine the mechanics of atoms and thus molecules. However, the solution to this equation for complicated potentials and for molecules with many electrons is very difficult and for many systems it has not been solved as of yet. Thus approximations must be made, and MD was created. The first step, the Born-Oppenheimer approximation, is to assume a solution to the SE that separates the solutions into one part for the electrons, and one part for the nuclei. Thus for each set of nuclear positions, we can solve the electron part of the SE, and find the electronic energy. This is known as the inter-atomic potential. Combining this with the nuclear-nuclear repulsions from the other part of the approximation we can find the total potential energy of the system. We now have an energy that is a function of atomic positions, and we call this the potential energy surface. Next we treat the system of nuclei as classical particles moving along this potential energy surface. Newton's equations of motion can now be used. The last approximation is to assume an analytic function that describes the potential energy surface [1].

To perform an MD simulation several components of Statistical Mechanics need to be taken into account. First, a thermostat is needed to control the temperature of the system. There are several types of thermostats that can be used, but the one that has been proven to work the best, is a combination of the Nose and Hoover thermostats [1]. This couples the system to a heat bath using a fictional dynamical variable in the equations of motion. Moldy gives the user three

options of thermostats. One is a simple scaling parameter, which rescales the system every few timesteps, so that the temperature remains constant. The two others are thermostats that couple the system to a heat bath as described above. These are the Gaussian thermostat and the Nose-Hoover thermostat. Secondly, an integrator is needed to numerically solve Newton's Equations. The typical method is to Taylor-expand the equations of motion with some alterations. Moldy uses a modified version of the Beeman algorithm to integrate [3]. Thirdly, a potential energy function is needed to calculate the forces acting on the system. This is the function that would come from the Born-Oppenheimer approximation. For large systems of atoms or molecules, this is usually divided into an electrostatic and an inter-molecular potential. For the inter-molecular one, there are several to choose from depending on what type of system one is simulating. A typical function is a combination of the nuclear repulsion and valence electron attractions. The electrostatic part is handled using the Ewald Sum. This sum modifies the basic concept that the system is of charged point ions mutually interacting via the Coulomb potential. The first modification is that each ion is neutralized at a long range by the superposition of a spherical gaussian cloud of opposite charge centered on each ion. This combination of point ions and gaussian charges make up the real-part term of the Ewald Sum. The second modification is to superimpose a second gaussian cloud with the same charge as the original point ions, thus nullifying the effects of the first cloud (i.e. we are left with the original system). Poisson's equation is used to calculate this effect in Fourier or reciprocal space. Additional terms such as a self-energy correction due to the fictitious presence of the gaussian clouds are added as well [2]. With these potential functions, one can compute the forces acting on the system, and then numerically solve Newton's Equations of Motion.

**Setting up a Moldy Run (Argon as example)**

To use Moldy for your molecular dynamic simulations two files need to be created. The first is what is called the control file. This file contains a list of parameters, which contain information on how to run the specific simulation. These include various thermodynamic parameters such as temperature, as well as statistical parameters such as how long to run the simulation, how often to calculate data, etc. A full list of all the parameters is appended to this paper [Table 1]. The most important parameter is a link to the file known as the specification file. This file contains information about the position of each atom, the types of atoms, the mass, charge, and if it is a solid the lattice structure of the system. It also identifies which inter-molecular potential will be used and gives the parameters of this potential. An example of a control file and a specification file of Argon gas is below:

```
# Control file for Argon at 84K
sys-spec-file=argon.in
density=1.428
temperature=84
scale-interval=5
scale-end=2500
nsteps=5000
print-interval=500
roll-interval=500
begin-average=2501
average-interval=2500
step=0.01
subcell=2
strict-cutoff=1
```

```
cutoff=8.5125  #2.5 * sigma
begin-rdf=2501
rdf-interval=50
rdf-out=2500
```

This file states that the specification file is named argon.in. The density of the system is 1.428 g/cm$^3$ . The density parameter is used only when the system does not have well-defined lattice structure, such as in a gas. The temperature is 84K, and is controlled by a scaling every 5 timesteps, but the temperature is allowed to vary after 2500 timesteps. There are a total of 5000 timesteps, data is printed every 500 timesteps as well as rolling averages of this data. Total averages are calculated beginning at the 2501st timestep and are averaged over the next 2500 timesteps (i.e. until the end of the run). Each timestep is 0.01 pico seconds, meaning the entire run is 0.01 * 5000 = 50ps. No actual experiment could calculate data on this scale and for this short of a time period. Thus, MD is necessary to study local properties of molecules at instantaneous moments.

The subcell size equals 2 angstrom. When modeling large systems of molecules, computer time becomes more and more critical. Thus, in order to simplify calculations molecules are broken up into unit cells or MD cells, which are repeated off into 3 directions, creating a large periodic system. Calculations made on each MD cell are then extended to the rest of the system. Each MD cell is further broken into subcells. Moldy uses the concept of the link cell method when making calculations. Since the force laws are inverse powers of radial distance, a cutoff radius can be chosen for each atom. In other words, interactions between atoms that are far apart are negligible, and thus are ignored in calculations. This cutoff radius depends on the pair-pair potential being used. It is common practice to use a cutoff radius such that the integral of the potential up to that cutoff radius is about 96 percent of the integral up to infinity. Moldy calculates interactions between atoms that are contained in subcells, which fall within the cutoff radius of one another. There are two options to choose from. Parameter strict-cutoff = 0 allows calculations of all subcells whose center lies within the cutoff. Strict-cutoff = 1 allows calculations of all subcells with any part lying within the cutoff [3]. In this example, all subcells are included which contain any part within 8.5125 angstroms of the given atom. Finally, radial distribution functions will be calculated starting at timestep 2501, calculated every 50 timesteps, and the data is printed after 2500 timesteps.

```
# Specification file for Argon using Lennard-Jones Potential
# Parameters from Allen and Tildesley Table 1.1 [4]
Argon 108
1       0       0       0   39.948 0 Ar
end
Lennard-Jones
1 1 3.984 3.41
```

This file states that there are 108 atoms of Argon. The first number under the title refers to the site-id of the atom. Since there is only one type of atom, only one id is necessary. This is followed by the x,y,z coordinates of the system. Again since there is only one atom, one is placed at the origin and then periodically repeated in all 3 directions. The mass in amu is given next, followed by the charge, and the name of the specific atom.
Underneath this is the label of the potential energy function used. Moldy contains 6 functions that can be input simply by plugging in the parameters of the function. Additional functions can

be added by changing the Moldy code itself. (See manual for instructions). The first two values under the title of the potential refer to the two sites that are interacting. Thus, the Lennard Jones potential is used between sites 1 and 1, or between any two Argon atoms in this case. Then the values of the parameters of the potential function are inserted. These vary from the type of atoms being used. The Moldy manual will describe what order to put in the parameters.
If a solid structure is going to be created and a lattice start used then the crystal structure must be defined at the bottom of the of the specification file as follows:

```
3.6     3.6     3.6     90      90      90      3       3       3
copper  0       0       0
copper  0.5     0.5     0
copper  0.5     0       0.5
copper  0       0.5     0.5
```

The first three values correspond to the length of each side of the MD cell, or the lattice parameter. The next three correspond to the angle between each MD cell side. In this case all three sides are equal and are perpendicular. Thus a cube is produced. The final 3 values on the top line describe how many cells should be repeated out in the direction of each MD cell vector. Underneath this is a description of each atom and its position. Only a certain number are needed since the cell will be repeated outward.

Assuming the Moldy code has been properly compiled, and after creating the two above files, a simulation can be run. Using a DOS format, this is executed by typing

moldy control.argon argon.out

at the prompt. This runs the file control.argon, and sends the results to the file argon.out. What is produced is a banner page summarizing most of the parameters of the two input files, a periodic output of thermodynamic properties, a final total average of these properties, and a list of radial distribution function values.

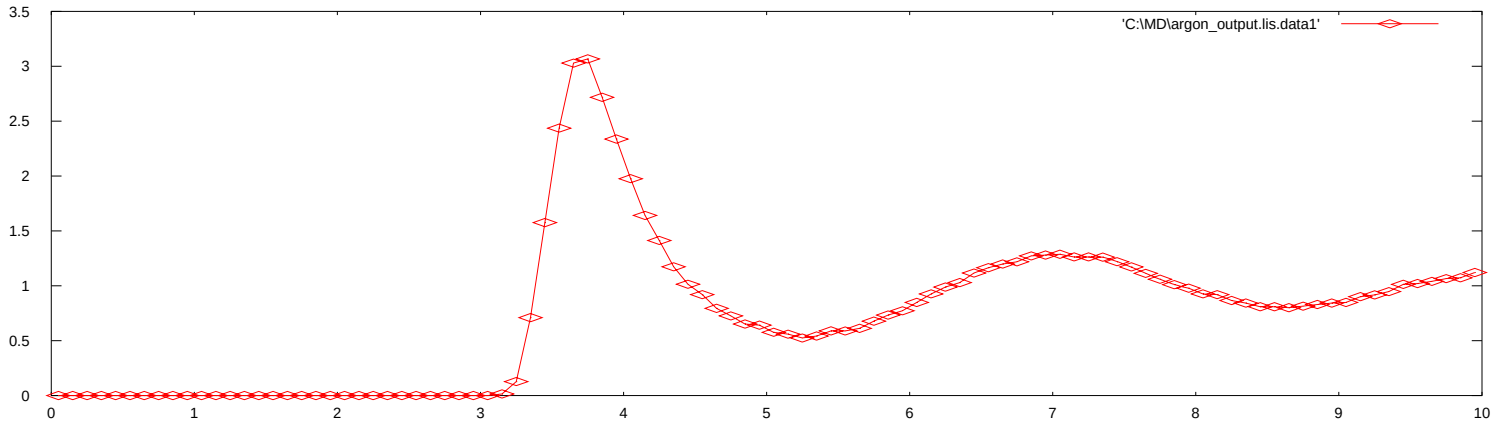
An example of the periodic output:

```
 Trans KE     Rot KE  Pot Energy  Tot Energy  TTemp  RTemp   Temp h(1,*) h(2,*) h(3,*)    Stress    Stress    Stress
======== Timestep 500     Current values
==================================================================================
    114.61        0       -661.46    -546.85        85.1   0.0    85.9  17.12   0.00   0.00    -11.6     4.74   0.00517
                                0                                        0.00  17.12   0.00     4.74     6.53     0.439
                                                                         0.00   0.00  17.12   0.00517    0.439      45.6
-------- Rolling averages over last 500 timesteps ---------------------------------------------------------------------
    112.62        0       -668.07    -555.46        83.6   0.0    84.4  17.12   0.00   0.00     1.88    -13.6      1.92
                                0                                        0.00  17.12   0.00    -13.6    -1.63     -23.1
                                                                         0.00   0.00  17.12     1.92    -23.1     -10.4
-------- Standard deviations -------------------------------------------------------------------------------------------
    3.6427        0        11.857     13.496         2.7   0.0     2.7   0.00   0.00   0.00     25.5     13.4      18.4
                                0                                        0.00   0.00   0.00     13.4     41.1      22.1
                                                                         0.00   0.00   0.00     18.4     22.1      34.6
```

This is output from the 500th timestep. The first section gives the instantaneous data, the second gives the rolling averages over the last 500 timesteps (specified in the control file), and the third gives the standard deviations of these averages. The last two bits of data are the MD cell (h) and

stress 3x3 matrices. To simulate bulk systems Moldy employs periodic boundary conditions. The program periodically repeats a unit cell of volume V containing N particles. Thus a molecular system consists of N particles in a cell of volume V that is periodically repeated to fill all space. This MD cell can have arbitrary shape and volume and can be completely described by three vectors: a, b, and c, which are the columns of the h matrix. These vectors span the edges of the MD cell, and can have different lengths and arbitrary orientations with a few rules. For one the first vector a, is confined to lie along the x-axis. Secondly, the vector b, is confined to lie in the x-y plane. These rules are used to avoid the phenomena of superfluous rotations that result from degrees of freedom that simply do not exist. A 3x3 matrix has nine degrees of freedom, but only six are needed to describe a 3-D molecule. Thus the MD cell can be thought of as sitting on a flat area under the influence of say a weak gravitational field. The volume of this MD cell can be calculated simply by taking the determinant of the h-matrix (equivalent to taking the triple product of the three vectors) [5]. In this case, the MD cell is a perfect cube, with each side being 17.12 angstroms long. The stress matrix is the standard stress matrix used in statistical mechanics.

The total average portion gives similar data but averages it over a longer time period. The final component of the output is the radial distribution function (RDF) output. The RDF gives a probability that for any given distance you will find two atoms that far apart. The RDF data is printed as list of values starting at r=0.1 angstroms up to ten angstroms. In order to view this data better, Moldy has a utility program, plotrdf, which takes the RDF data and makes it readable in a graphing program. The script program Perl must be installed on your computer as well as a graphing program such as Gnuplot. The resulting RDF curve for this argon example follows:

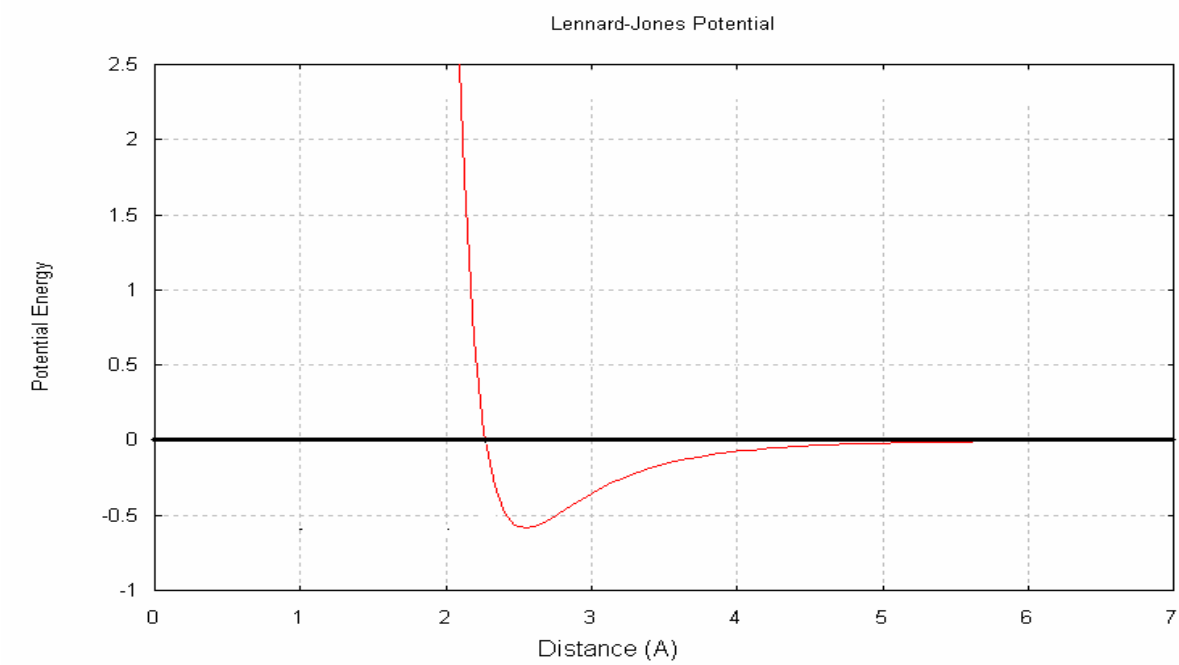



The x-axis goes from 0 to 10 angstroms, and is intervaled into bins of 0.1 angstroms. According to this data, most of the atoms will be a distance of 3.8 angstroms. After this it appears that there is equal probability of atoms being any distance apart. This is due to the fact that Argon is a gas. When we look at the data from copper, a solid crystal structure, we will see many well-defined peaks.

**Creating the Copper Crystal**

The first step in creating a copper crystal was to find the correct parameters for the control file. Several control files were made. One was made with 5000 timesteps and one was made with 100,000 timesteps. In both cases, each timestep was 0.005 picoseconds long. It was discovered that the 100,000 timestep runs had data that converged more to an equilibrium position, due to the fact that it had a longer time to settle. These runs, however, took up a more substantial amount of computing time. In all cases the total averages and RDF values were calculated for the last half of the total run. Simulations were performed at temperatures of 300K and 77K. In most cases, we used the scaling option to control the temperature of the system. When we did perform a run using the Gaussian thermostat, the only difference we found was in the standard deviations of the average temperatures. Using the thermostat seemed to create a more stable system, with smaller temperature fluctuations. Also the parameter time-unit was fixed at 1.0181e-14. This ensured that all values in the specification file were in eV, Angstroms, and amu.

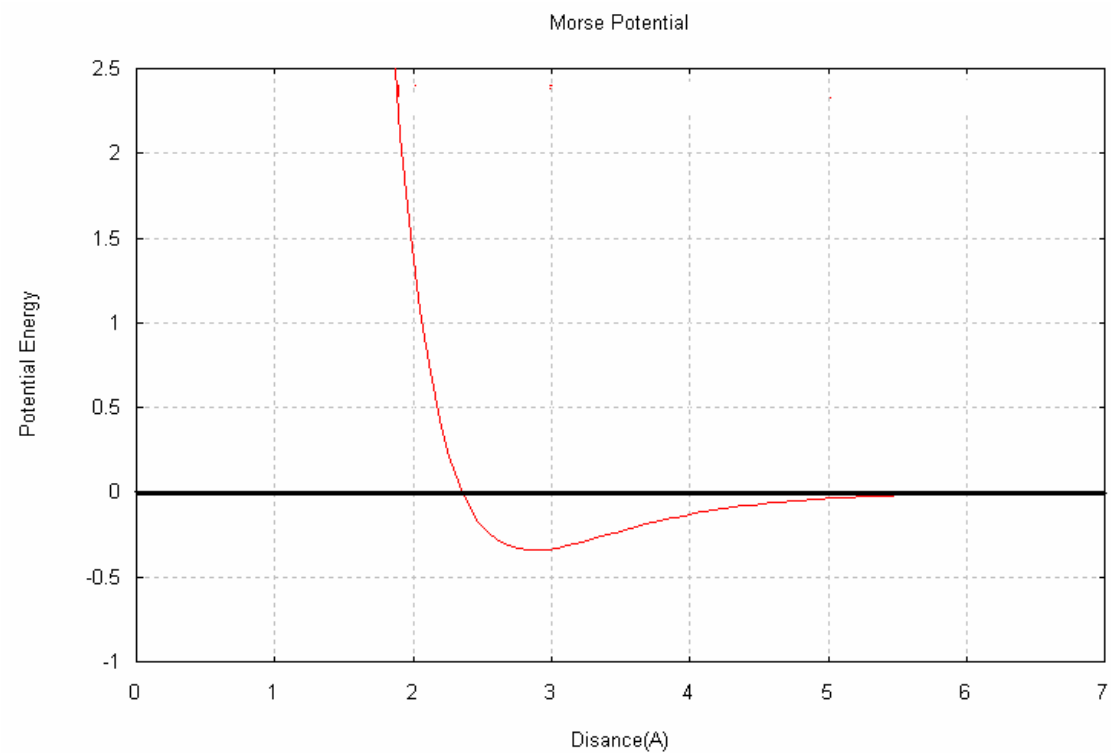
One of the first challenges was choosing an accurate cutoff radius. First, however, an appropriate potential energy function was needed, and thus a specification file needed to be created. Since we were simulating a solid, a lattice start was used, and the crystal structure was defined in the specification file as well. This made it possible to view the molecule in a visual graphing program, Visual Molecular Dynamics (VMD). After reading some literature, we realized that the ideal potential to use for a copper crystal would be a many-body potential. This not only takes into account the pair-pair interactions between each atom, but the interactions between each atom and the system as a whole. One such potential, the Embedded-Atom-Method (EAM), was found to be useful for simulating copper [6]. However, it was discovered that the Moldy program is not enabled to perform calculations of many-body potentials. Only simulations with pair-pair potentials can be used. While one route would be to modify the Moldy code itself, so that it could handle the EAM potential, we chose to run simulations using two types of simple pair-pair potentials. Taking the Argon gas as an example, we made some simulations of copper using the Lennard-Jones potential:



$$4\varepsilon\,[(\sigma/r)^{12} - (\sigma/r)^{6}] \qquad [7]$$

Here $\varepsilon$ = minimum value of the potential and $\sigma$ equal the first zero of the potential: $V(\sigma) = 0$.
r = the distance between the two atoms whose interaction is being calculated at the time

The second potential we used was the Morse Potential,




$D\,[e^{\{-2\alpha(r_{ij} - r_o)\}} - 2\,e^{\{-\alpha\,(r_{ij} - r_o)\}}]$,

$r_{ij}$ – Distance between atoms i and j
D – Disassociation energy [eV]
$r_o$ – Equilibrium distance [Angstroms]
$\alpha$ – [1/Angstrom]

This form of the Morse potential had been used in uniaxial tension simulations of pure copper and copper alloys [8]. Thus, it was known that it should work well.

As one can see the two potentials are very similar, and in fact produced identical results. The only difference came in computation time, where the Morse potential runs took more time due to the fact that it needed more complicated computations. The parameters used in the specification file were obtained from literature [7,8], and can be found in the specification files attached to the end of this paper.

Now that a potential function was chosen, a cutoff radius could be calculated. This was

done by the integral method mentioned above. For the Lennard-Jones potential, a radius of 4.88 angstrom was calculated. Since the Morse potential was very similar, this value was also used. In addition to this, a subcell size of 2 angstrom was used, and we used the cutoff method that included all subcells with any part within the cutoff radius. This made sure that any interaction that may have some effect was taken into account.

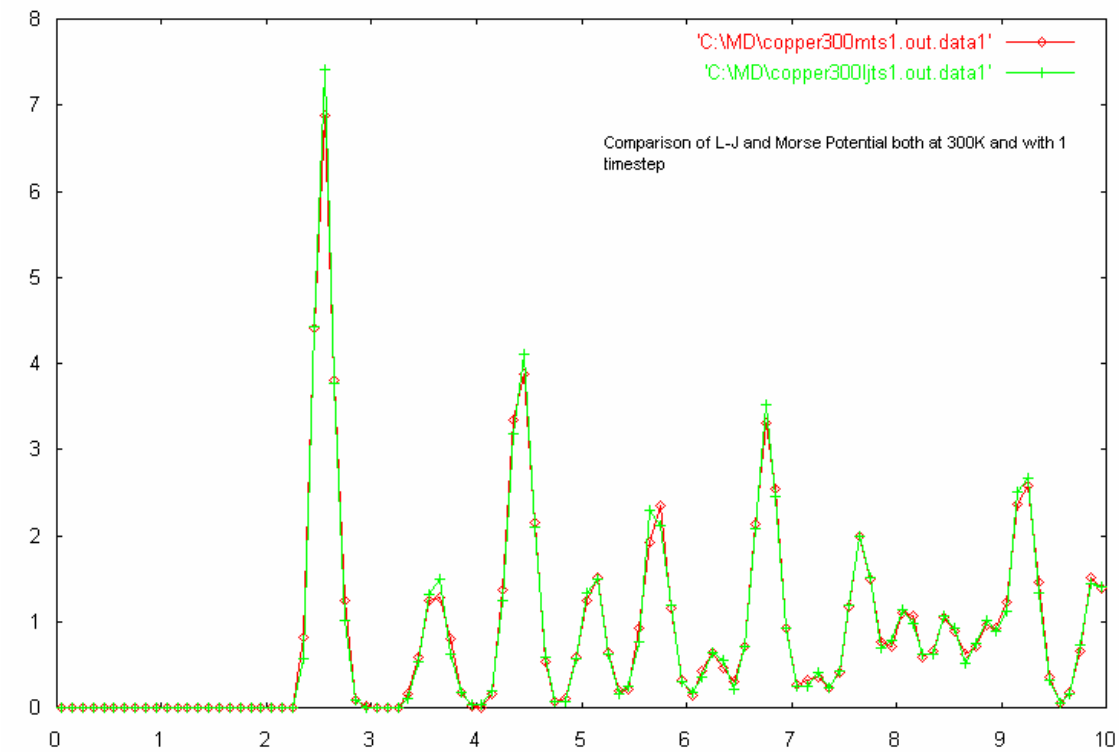
The next task was to create the crystal structure within the specification file. For both potentials the structure of the copper crystal was the same. 108 atoms of Copper were simulated. The lattice parameter of Copper, 0.36149nm, was used as the length of each side of the unit cell [9]. In order to make a cube, an angle of 90 degrees between each side was used. Finally to create an FCC structure, we used four atoms as the building blocks. Since the unit cells will be built up on top of each other, several cells will share the same atoms. Thus, only four atoms were necessary. Eight unit cells would share the one at the origin, and all corners of the structure would be modeled. Two cells would share each face-oriented atom. Thus, three of these were needed; one in the x-y plane, one in the x-z plane, and one in the y-z plane.

While Moldy has this feature that only four atoms were needed to build an entire FCC structure, we found that it might be useful to create a program that produces the structure by identifying each atom. This would allow for the input of dislocations or gaps within the crystal structure, and therefore non-perfect (more realistic) crystals could be produced. A Matlab program was created that could build either an FCC or a BCC molecule of one type of atom, such that a gap along an arbitrary plane could be produced. The output of this program was identical to what would be input into the specification file for each atom. A simulation of a 3x3x3 FCC copper molecule with a gap at the x=1.5 plane was produced (each unit cell having length of 1).

**Output and Conclusions**

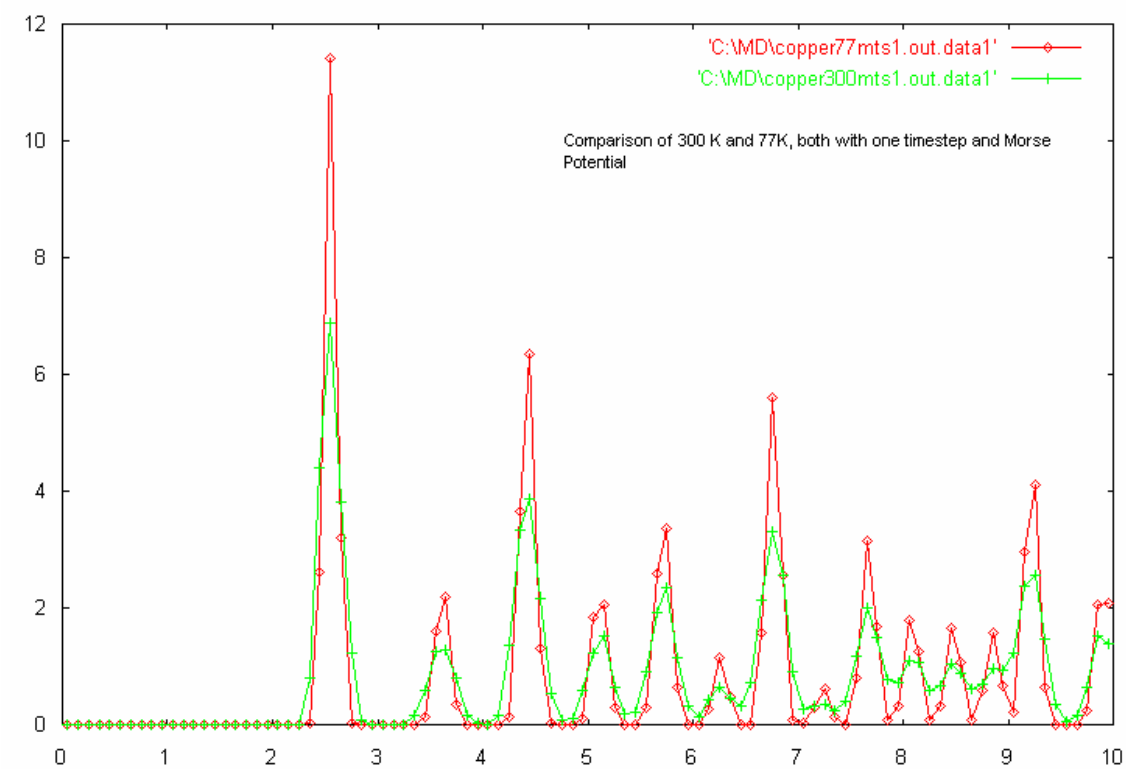
For each run, Moldy created an output file, which contained the thermodynamic properties of the copper crystal. Values were calculated instantaneously at each time-step, averaged over each time-step, and averaged over the last half of the total run. Moldy also produced a radial distribution curve for each run. Finally, we used the graphics program VMD to actually view the crystal.

In comparing the Morse and the Lennard-Jones potentials, both produced strikingly similar results. As can be seen in the two RDF curves:



Here the green line is the Lennard-Jones run and the red is the Morse run.

The difference between the runs with many time-steps and only one were negligible. The real results came when comparing the simulations of different temperatures. The differences between runs at different temperatures occurred in the Kinetic Energy of the systems and the RDF curves. Of course for the 77K run, the KE was much smaller. The RDF curve, though, had narrower peaks, and attained higher values than the 300K run. The reason for the narrower peaks at lower temperature can be explained by the fact that at a lower temperature the atoms are going to be moving at lower speeds and have fewer vibrations. Thus the atoms will be more confined to a particular position. These positions or peaks, can also be explained. The first peak, which happens to be the largest, corresponds to the nearest neighbor distance (i.e. the distance at which each atom is closest to another). In the FCC structure, this corresponds to the distance between a corner atom and a face atom on the same plane. The next peak refers to the lattice parameter, which is the distance from a corner atom to another corner along a one-dimensional direction. This is followed by a peak at a distance between a face atom and a corner atom on separate planes, followed by twice the nearest neighbor distance, and then twice the lattice parameter. Several peaks, which correspond to the distance between certain atoms in the FCC structure follow, and can all be explained geometrically. This result alone helped us become confident that we had indeed created an FCC structured molecule.




Next we wanted to simulate the effects of thermal expansion and contraction. However, for a temperature difference of 300K to 77K, the contraction would be too small and the precision of the RDF values would not be small enough to notice the difference. Using linear thermal expansion as an approximation for the change in length of the lattice parameter, we calculated what the difference in length would be for a temperature change of 77K to 300K.

Using:

$\Delta L = \alpha L_o \Delta T$

We calculated that the change in lattice parameter would only be 0.009 angstroms. Even with changing the number of bins to calculate RDFs to 1000 this only could show differences of 0.01. So we decided to compare the curves for a run at 1K and a run at 900K. This should lead to a lattice length change of 0.03 angstroms, which could be noticed by taking the weighted averages of RDF values around each peak. Sure enough, the thermal expansion was detected. Therefore, the Moldy program was taking into account temperature and the motion of the atoms when building the copper crystals.

As for the visual simulations, using VMD, we are pretty confident that it created the proper FCC structure. And for both potentials, the exact results were obtained. In addition the simulation of the copper structure with the gap at x = 1.5 showed that the gap changed the orientation of the system to a lower atomic plane. This was further evidence that the Moldy program was producing results that were consistent with experimental data. It also gave us insights into what exactly was happening locally to each atom at different temperatures during very short intervals, which could not be seen experimentally. By comparing Argon and Copper we were also able to see more precise differences between a gas structure and a solid crystal one.

**Future**

More simulations of copper are needed to make sure that the data that Moldy is

producing is accurate. In addition, finding more potentials that better relate to interactions within copper molecules need to be discovered. One future task, would be to modify the Moldy base code, and allow it to use many-body potentials such as EAM. This would hopefully produce more realistic and accurate simulations.

Using the Matlab program to create various forms of the copper crystal, we can then study various properties of copper. One important application would be in the study of dislocations. Here at the National High Magnetic Field Laboratory (NHMFL), copper is used as conducting material in magnets, due to its high conductivity. However, it is a very weak material. High-powered magnets put a lot of stress on the materials within themselves due to the high Lorentz forces. When trying to increase the strength of copper, though, by rolling or bending, the material loses some of its conductivity. Therefore, ways of increasing copper's strength without losing too much of its conductivity need to be studied. Studies of nano-indentation and dislocations are being done in order to see if this can be achieved. It is hoped that the Moldy program can be used to simulate the effects of nano-indentation and dislocations on the conductivity and the strength of copper and copper alloys.

**Acknowledgements**

The author would like to thank the National High Magnetic Field Laboratory in Tallahassee, Florida and the National Science Foundation for making this research possible. Special thanks to Dr. Ke Han and Mr. Jingping Chen for mentoring this research project and for all their wisdom and patience, and to the Center for Integrating Research and Learning staff for all their personal support.

**References**


**[1]** http://www.ch.embnet.org/MD_tutorial/

**[2]** http://www.dl.ac.uk/TCS/Software/DL_POLY/USRMAN/node64.html

**[3]** Refson, Keith Moldy User's Manual, Revision 2.26.2.6, May 24, 2001

**[4]** M. P. Allen and D. J. Tildesley, Computer simulation of liquids, Clarendon Press, Oxford, 1987.

**[5]** M. Parrinello and A. Rahman, Phys. Rev. Lett. 45 (1980) p. 1196

**[6]** S. M. Foiles, M. I. Baskes, and M. S. Daw, Phys. Rev. B 33 num 12 (1986) p. 7983

**[7]** http://web.mse.uiuc.edu/matse390/projects/2001/team5/node13.html
Department of Materials Science: University of Illinois, Urbana-Champaign

**[8]** Komanduri, R., Chandrasekaran, N., Raff, L. M. Molecular Dynamics Simulation of Uniaxial Tension of Some Single-Crystal Cubic Metals at Nanolevel. International Journal of Mechanical Sciences: 43; (2001) p. 2252

**[9]** http://www.webelements.com

## Table 1.1 Control Parameters [3]

| Name | Type | Default | Function |
|---|---|---|---|
| Title | character | Test Simulation | A title to be printed on all output |
| nsteps | integer | 0 | Number of MD steps to execute |
| cpu-limit | real | 1e20 | Terminate run if excessive CPU time used |
| step | real | 0.005 | Size of timestep (in ps) |
| sys-spec-file | character | Null | Name of system specification file. Appended to control file if null |
| lattice-start | Boolean | False | Switch for crystalline initial configuration. |
| save-file | Character | Null | File to save restart configuration in. |
| restart-file | Character | Null | File to read restart configuration from. |
| new-sys-spec | Boolean | False | Read restart configuration with changed system specification |
| text-mode-save | Boolean | False | Write a portable “restart” file consisting of control, system specification and lattice start files. |
| Density | Real | 1.0 | Initial density in g/cm$^3$. Used by skew start only to determine initial MD cell dimensions |
| scale-interval | Integer | 10 | Number of steps between velocity scalings. |
| scale-end | Integer | 1000000 | When to stop scaling. |
| const-temp | Integer | 0 | 1 for Nose-Hoover, 2 for Gaussian thermostat. |
| ttmass | Real | 100 | Translational inertia parameter for Nose-Hoover thermostat (kJ/mol ps^2) |
| rtmass | Real | 100 | Rotational inertia parameter for Nose-Hoover thermostat |
| scale-options | Integer | 0 | Select variations on scaling or thermostat. |
| temperature | Real | 0 | Temperature of initial configuration for scaling and thermostat (K) |
| const-pressure | Integer | 0 | 1 Parrinello and Rahman constant stress<br>2 P-R compatible constant pressure ensemble<br>3 Wentzcovitch/Cleveland constant stress<br>4 Anderson constant pressure. |
| w | Real | 100.0 | Extended system mass parameter (amu) |
| pressure | Real | 0 | External applied pressure (MPa). |
| strain-mask | Integer | 200 | Bitmask controlling h matrix constraint. |
| Alpha | Real | Auto | $\alpha$ parameter from Ewald sum. Set any negative value to disable Ewald sum. |
| k-cutoff | Real | Auto | Reciprocal space cutoff distance |
| cutoff | Real | Auto | Direct space cutoff distance |
| strict-cutoff | Boolean | False | Flag to select rigorous or cheap but approximate cutoff algorithm |
| surface dipole | Boolean | False | Include surface dipole term in Ewald sum. |
| roll-interval | Integer | 10 | Period over which to calculate rolling averages |
| print-interval | Integer | 10 | How frequently to print normal output. |
| begin-average | Integer | 1001 | When to start accumulating thermodynamic averages |
| average-interval | Integer | 5000 | How frequently to calculate and print averages |
| reset-averages | Boolean | false | Discard accumulated averages in restart file. |
| begin-rdf | Integer | 1000000 | When to start accumulating radial distribution function information |
| rdf-interval | Integer | 20 | How frequently binning calculation is performed |
| rdf-out | Integer | 5000 | How frequently to calculate and print RDFs. |
| rdf-limit | Real | 10 | Calculate RDFs up to what distance (A) |
| nbins | Integer | 100 | Number of binning intervals between 0 and rdf-limit |
| xdr | Boolean | True | Write restart, backup and dump files in portable binary format using Sun XDR |

| | | | |
|---|---|---|---|
| dump-file<br>begin-dump<br>dump-interval<br>dump-level<br>ndumps | Character<br>Integer<br>Integer<br>Integer<br>Integer | Null<br>1<br>20<br>0<br>250 | Template of file names used for data dumps<br>Timestep to begin dumping at<br>How frequently to perform dumps<br>Amount of information to include in dump.<br>Number of dump records in each dump file. |
| backup-interval<br>backup-file | Integer<br>Character | 500<br>MDBACKUP | Frequency to write backup file<br>Name of backup file. |
| temp-file | Character | MDTEMPX | Name of temporary file used for writing restart configurations. |
| Subcell | Real | 0 | Size of sub-cell (in A) to divide MD cell into for link cell force calculation. |
| Seed | Integer | 1234567 | Seed for random number generator |
| page-width<br>page-length | Integer<br>Integer | 132<br>44 | Number of columns on output paper<br>Number of lines on page output |
| mass-unit<br>length-unit<br>time-unit<br>charge-unit | Real<br>Real<br>Real<br>Real | 1.6605402e-27<br>1e-10<br>1e-13<br>1.60217733e-19 | Unit of mass for system specification file.<br>Unit of length for system specification file.<br>Unit of time for system specification file.<br>Unit of charge for system specification file. |
| | | | |

## Examples of Copper control files

```
# Control file for copper crystal at 300 K
# Used with the Lennard-Jones Potential
# Accurate cutoff

title=Copper Crystal 300K L-J rc=4.88
sys-spec-file=copperLJ.in
nsteps=5000
step=0.005
lattice-start=1
print-interval=1000
temperature=300
strict-cutoff=1
roll-interval=1000
begin-average=2501
average-interval=2500
begin-rdf=2501
rdf-interval=50
rdf-out=2500
cutoff=4.88
subcell=2
time-unit=1.0181e-14
end
```

```
# Control file for copper crystal at 77 K
# Used with the Morse Potential
# 1 timestep

title=Copper Crystal 77K Morse 1 timestep
sys-spec-file=copperM.in
nsteps=5000
step=0.005
lattice-start=1
print-interval=5000
temperature=77
strict-cutoff=1
roll-interval=5000
begin-average=2501
average-interval=2500
begin-rdf=2501
rdf-interval=2500
rdf-out=2500
cutoff=4.88
subcell=2
time-unit=1.0181e-14
end
```

## Examples of Copper Specification Files

```
# Crystal Structure and Potential parameters for Copper
# Using Lennard-Jones Potential
# Unit cell parameters from http://www.webelements.com

Copper 108
1       0       0       0       63.546  0       Cu
end
Lennard-Jones
1 1 0.5838 2.27
end
3.6149  3.6149  3.6149  90  90  90 3  3  3
Copper 0       0       0
Copper 0.5     0       0.5
Copper 0.5     0.5     0
Copper 0       0.5     0.5
end
```

```
# Crystal Structure and Potential parameters for Copper
# Using Morse Potential
# Unit cell parameters from http://www.webelements.com

Copper 108
1	0	0	0	63.546	0	Cu
end
morse
1 1 0 0  0 0 0.3429 1.3558 2.626
end
3.6149  3.6149  3.6149  90  90  90 3  3  3
Copper 0	0	0
Copper 0.5	0	0.5
Copper 0.5	0.5	0
Copper 0	0.5	0.5
end
```

```
# Crystal Structure and Potential for Copper
# Using Lennard Jones Potential
# Unit cells from Matlab code
# FCC structure 3x3x3
# Lattice parameter = (3+1)*lattice
# Gap placed along plane where x = 1.5

copper 148
1	0	0	0	63.546	0	Cu
end
Lennard-Jones
1 1 0.5838 2.27
end
14.46	14.46	14.46	90	90	90	1	1	1
copper	0	0	0
copper	0	0.25	0
copper	0	0.5	0
copper	0	0.75	0
copper	0.125	0.125	0
copper	0.125	0.375	0
copper	0.125	0.625	0
copper	0.25	0	0
copper	0.25	0.25	0
copper	0.25	0.5	0
copper	0.25	0.75	0
copper	0.5	0	0
copper	0.5	0.25	0
copper	0.5	0.5	0
copper	0.5	0.75	0
copper	0.625	0.125	0
copper	0.625	0.375	0
```

| | | | |
|---|---|---|---|
| copper | 0.625 | 0.625 | 0 |
| copper | 0.75 | 0 | 0 |
| copper | 0.75 | 0.25 | 0 |
| copper | 0.75 | 0.5 | 0 |
| copper | 0.75 | 0.75 | 0 |
| copper | 0 | 0.125 | 0.125 |
| copper | 0 | 0.375 | 0.125 |
| copper | 0 | 0.625 | 0.125 |
| copper | 0.125 | 0 | 0.125 |
| copper | 0.125 | 0.25 | 0.125 |
| copper | 0.125 | 0.5 | 0.125 |
| copper | 0.125 | 0.75 | 0.125 |
| copper | 0.25 | 0.125 | 0.125 |
| copper | 0.25 | 0.375 | 0.125 |
| copper | 0.25 | 0.625 | 0.125 |
| copper | 0.5 | 0.125 | 0.125 |
| copper | 0.5 | 0.375 | 0.125 |
| copper | 0.5 | 0.625 | 0.125 |
| copper | 0.625 | 0 | 0.125 |
| copper | 0.625 | 0.25 | 0.125 |
| copper | 0.625 | 0.5 | 0.125 |
| copper | 0.625 | 0.75 | 0.125 |
| copper | 0.75 | 0.125 | 0.125 |
| copper | 0.75 | 0.375 | 0.125 |
| copper | 0.75 | 0.625 | 0.125 |
| copper | 0 | 0 | 0.25 |
| copper | 0 | 0.25 | 0.25 |
| copper | 0 | 0.5 | 0.25 |
| copper | 0 | 0.75 | 0.25 |
| copper | 0.125 | 0.125 | 0.25 |
| copper | 0.125 | 0.375 | 0.25 |
| copper | 0.125 | 0.625 | 0.25 |
| copper | 0.25 | 0 | 0.25 |
| copper | 0.25 | 0.25 | 0.25 |
| copper | 0.25 | 0.5 | 0.25 |
| copper | 0.25 | 0.75 | 0.25 |
| copper | 0.5 | 0 | 0.25 |
| copper | 0.5 | 0.25 | 0.25 |
| copper | 0.5 | 0.5 | 0.25 |
| copper | 0.5 | 0.75 | 0.25 |
| copper | 0.625 | 0.125 | 0.25 |
| copper | 0.625 | 0.375 | 0.25 |
| copper | 0.625 | 0.625 | 0.25 |
| copper | 0.75 | 0 | 0.25 |
| copper | 0.75 | 0.25 | 0.25 |
| copper | 0.75 | 0.5 | 0.25 |
| copper | 0.75 | 0.75 | 0.25 |
| copper | 0 | 0.125 | 0.375 |
| copper | 0 | 0.375 | 0.375 |
| copper | 0 | 0.625 | 0.375 |
| copper | 0.125 | 0 | 0.375 |
| copper | 0.125 | 0.25 | 0.375 |
| copper | 0.125 | 0.5 | 0.375 |
| copper | 0.125 | 0.75 | 0.375 |
| copper | 0.25 | 0.125 | 0.375 |
| copper | 0.25 | 0.375 | 0.375 |

| | | | |
|---|---|---|---|
| copper | 0.25 | 0.625 | 0.375 |
| copper | 0.5 | 0.125 | 0.375 |
| copper | 0.5 | 0.375 | 0.375 |
| copper | 0.5 | 0.625 | 0.375 |
| copper | 0.625 | 0 | 0.375 |
| copper | 0.625 | 0.25 | 0.375 |
| copper | 0.625 | 0.5 | 0.375 |
| copper | 0.625 | 0.75 | 0.375 |
| copper | 0.75 | 0.125 | 0.375 |
| copper | 0.75 | 0.375 | 0.375 |
| copper | 0.75 | 0.625 | 0.375 |
| copper | 0 | 0 | 0.5 |
| copper | 0 | 0.25 | 0.5 |
| copper | 0 | 0.5 | 0.5 |
| copper | 0 | 0.75 | 0.5 |
| copper | 0.125 | 0.125 | 0.5 |
| copper | 0.125 | 0.375 | 0.5 |
| copper | 0.125 | 0.625 | 0.5 |
| copper | 0.25 | 0 | 0.5 |
| copper | 0.25 | 0.25 | 0.5 |
| copper | 0.25 | 0.5 | 0.5 |
| copper | 0.25 | 0.75 | 0.5 |
| copper | 0.5 | 0 | 0.5 |
| copper | 0.5 | 0.25 | 0.5 |
| copper | 0.5 | 0.5 | 0.5 |
| copper | 0.5 | 0.75 | 0.5 |
| copper | 0.625 | 0.125 | 0.5 |
| copper | 0.625 | 0.375 | 0.5 |
| copper | 0.625 | 0.625 | 0.5 |
| copper | 0.75 | 0 | 0.5 |
| copper | 0.75 | 0.25 | 0.5 |
| copper | 0.75 | 0.5 | 0.5 |
| copper | 0.75 | 0.75 | 0.5 |
| copper | 0 | 0.125 | 0.625 |
| copper | 0 | 0.375 | 0.625 |
| copper | 0 | 0.625 | 0.625 |
| copper | 0.125 | 0 | 0.625 |
| copper | 0.125 | 0.25 | 0.625 |
| copper | 0.125 | 0.5 | 0.625 |
| copper | 0.125 | 0.75 | 0.625 |
| copper | 0.25 | 0.125 | 0.625 |
| copper | 0.25 | 0.375 | 0.625 |
| copper | 0.25 | 0.625 | 0.625 |
| copper | 0.5 | 0.125 | 0.625 |
| copper | 0.5 | 0.375 | 0.625 |
| copper | 0.5 | 0.625 | 0.625 |
| copper | 0.625 | 0 | 0.625 |
| copper | 0.625 | 0.25 | 0.625 |
| copper | 0.625 | 0.5 | 0.625 |
| copper | 0.625 | 0.75 | 0.625 |
| copper | 0.75 | 0.125 | 0.625 |
| copper | 0.75 | 0.375 | 0.625 |
| copper | 0.75 | 0.625 | 0.625 |
| copper | 0 | 0 | 0.75 |
| copper | 0 | 0.25 | 0.75 |
| copper | 0 | 0.5 | 0.75 |

| | | | |
|---|---|---|---|
| copper | 0 | 0.75 | 0.75 |
| copper | 0.125 | 0.125 | 0.75 |
| copper | 0.125 | 0.375 | 0.75 |
| copper | 0.125 | 0.625 | 0.75 |
| copper | 0.25 | 0 | 0.75 |
| copper | 0.25 | 0.25 | 0.75 |
| copper | 0.25 | 0.5 | 0.75 |
| copper | 0.25 | 0.75 | 0.75 |
| copper | 0.5 | 0 | 0.75 |
| copper | 0.5 | 0.25 | 0.75 |
| copper | 0.5 | 0.5 | 0.75 |
| copper | 0.5 | 0.75 | 0.75 |
| copper | 0.625 | 0.125 | 0.75 |
| copper | 0.625 | 0.375 | 0.75 |
| copper | 0.625 | 0.625 | 0.75 |
| copper | 0.75 | 0 | 0.75 |
| copper | 0.75 | 0.25 | 0.75 |
| copper | 0.75 | 0.5 | 0.75 |
| copper | 0.75 | 0.75 | 0.75 |