1

A beginner's guide to the modeling of shock/uniaxial/quasi-isentropic compression using the LAMMPS molecular dynamics simulator By: Oscar Guerrero-Miramontes

The following text is a recompilation of LAMMPS [1] scripts that intends to explain in a nutshell the modeling of shock, uniaxial and quasi-isentropic compression using molecular dynamics simulations (MD). Hopefully these scripts may be useful for the MD newcomers. All the scripts provided in this text were tested using lammps-20Sep12 version.

There is no guarantee that the scripts are fail proof or will work properly with a more recent version.

1.1 Uniaxial compression in Copper (SCRIPT1):

The next script is useful when trying to model uniaxial compression in BCC/FCC crystals e.g Tantalum or Copper [2,3]. The uniaxial compression can be performed along different crystalographic directions: (100),(110) and (111) to name a few. The stress-strain curve and shear-strain curve are plotted using gnuplot and the elastic-plastic transition is visualized using ATOMEYE [4]. The atomic interactions are modeled using the embedded atom method (EAM) [5]

```
# Isothermal Uniaxial deformation along [100],[110] or [111]
# The equilibration step allows the lattice to expand to a
# temperature of 300 K with a pressure of 0 kbar at each si-
# mulation cell boundary. Then, the simulation cell is def-
# ormed in the x-direction at a strain rate of 0.005 1/ps,
# while the lateral boundaries are controlled using the NVT
# equations of motion to maintain the length of the computa-
# tional box constant. The stress and strain values are ou-
# tput to a separate file, which can be imported in a grap-
# hing application for plotting. The cfg dump files include
# the x, y, and z coordinates, the centrosymmetry values, the
# potential energies, and forces for each atom. This can be
# directly visualized using AtomEye. We assume 1 timestep is
# equal to 0.0005 pico-seconds

# Script made by Oscar Guerrero-Miramontes

# ---------- Initialize Simulation ---------------------
```

```
clear
units metal
dimension 3
boundary p p p
atom_style atomic
atom_modify map array

# ---------- define variables ---------------------

variable stemperature equal 300   # temperature in kelvin
variable epercentage  equal 0.30  # the percentage the body is compressed
variable myseed       equal 12345 # the value seed for the velocity
variable atomrate     equal 2500  # the rate in timestep that atoms are dump as CFG
variable time_step    equal 0.005 # time step in pico seconds
variable time_eq      equal 10000 # time steps for the equlibration part

variable tdamp equal "v_time_step*100"  # DO NOT CHANGE
variable pdamp equal "v_time_step*1000" # DO NOT CHANGE

variable R equal 0.005 # ERATE here 0.005 1/ps every picosecond

# time for the deformation part (DO NOT CHANGE)
variable time_run equal "(v_epercentage/v_R)/v_time_step"

timestep ${time_step} # DO NOT CHANGE

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

# Create FCC lattice with x orientation along (100)
# if you want to change orientation along (110) or (111)
# then uncomment

  lattice fcc 3.615 origin 0.0 0.0 0.0 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1
# lattice fcc 3.615 origin 0.0 0.0 0.0 orient x 1 1 0 orient y 0 0 1 orient z 1 -1 0
# lattice fcc 3.615 origin 0.0 0.0 0.0 orient x 1 1 1 orient y 1 -1 0 orient z 1 1 -2

region box block 0 50 0 50 0 50 units lattice
create_box 1 box
create_atoms 1 box

# ---------- Define Interatomic Potential ---------------------

pair_style eam/alloy
pair_coeff * * Cu_mishin1.eam.alloy Cu
```

```
# ---------- Define Settings ---------------------------------

# "compute" Define a computation that will be performed on a
# group of atoms. Quantities calculated by a compute are
# instantaneous values, meaning they are calculated from
# information about atoms on the current timestep or iteration,
# though a compute may internally store some information about
# a previous state of the system. Defining a compute does not
# perform a computation. Instead computes are invoked by other
# LAMMPS commands as needed

compute myCN all cna/atom 3.9
compute myKE all ke/atom
compute myPE all pe/atom

# ---------- Equilibration -------------------------------------

reset_timestep 0
velocity all create ${stemperature} ${myseed} mom yes rot no

# The equilibration step NPT allows the lattice to expand to a
# temperature of 300 K with a pressure of 0 bar at each simul-
# ation cell boundary. The Tdamp and Pdamp parameter is speci-
# fied in time units and determines how rapidly the temperatu-
# re is relaxed. A good choice for many models is a Tdamp of-
# around 100 timesteps and A good choice for many models is a-
# Pdamp of around 1000 timesteps. Note that this is NOT the s-
# ame as 1000 time units for most units settings.

fix equilibration all npt temp ${stemperature} ${stemperature} ${tdamp}
iso 0 0 ${pdamp} drag 1

# Output thermodynamics into outputfile
# for units metal, pressure is in [bars] = 100 [kPa] = 1/10000 [GPa] p2, p3, p4 are in GPa

variable eq1 equal "step"
variable eq2 equal "pxx/10000"
variable eq3 equal "pyy/10000"
variable eq4 equal "pzz/10000"
variable eq5 equal "lx"
variable eq6 equal "ly"
variable eq7 equal "lz"
variable eq8 equal "temp"
variable eq9 equal "etotal"
```

```
fix output1 all print 10 "${eq1} ${eq2} ${eq3} ${eq4} ${eq5} ${eq6} ${eq7} ${eq8} ${eq9}"
file run.${stemperature}K.out screen no

thermo 10
thermo_style custom step pxx pyy pzz lx ly lz temp etotal

# Use cfg for AtomEye (here only print the coordinates of the atoms)
dump 1 all cfg ${time_eq} fcc.cu.*.cfg id type xs ys zs
dump_modify 1 element Cu

# RUN AT LEAST 10000 timesteps
run ${time_eq}

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

# reset
unfix equilibration
undump 1
unfix output1

# ---------- Deformation ---------------------------------

reset_timestep 0

# In our simulations We seek to control the lateral boundaries
# Ly and Lz, i.e When the computational box is stretched,
# the contraction or transverse strain perpendicular to the
# load is zero

fix 1 all nvt temp ${stemperature} ${stemperature} ${tdamp} drag 0.0
fix 2 all deform 1 x erate -${R} units box remap x

# IMPORTANT NOTE: When non-equilibrium MD (NEMD) simulations are
# performed using this fix, the option "remap v" should normally
# be used. This is because fix nvt/sllod adjusts the atom positions
# and velocities to induce a velocity profile that matches the
# changing box size/shape. Thus atom coordinates should NOT be
# remapped by fix deform, but velocities SHOULD be when atoms cross
# periodic boundaries, since that is consistent with maintaining
# the velocity profile already created by fix nvt/sllod. LAMMPS
# will warn you if the remap setting is not consistent with fix nvt/sllod.
```

```
# Output strain and stress info to file
# for units metal, pressure is in [bars] = 100 [kPa] = 1/10000 #[GPa] pxx, pyy, pzz
# are in GPa

variable strain equal "-(lx - v_L0)/v_L0"
variable shear  equal "0.5*(pxx/10000 - 0.5*(pyy/10000 + pzz/10000))"
variable tstep  equal "step"
variable mypxx  equal "pxx/10000"
variable mypyy  equal "pyy/10000"
variable mypzz  equal "pzz/10000"
variable mylx   equal "lx"
variable myly   equal "ly"
variable mylz   equal "lz"
variable mytemp equal "temp"

fix def1 all print 10 "${strain} ${mypxx} ${mypyy} ${mypzz} ${shear} ${mylx} ${myly} ${mylz}
 ${mytemp}" file stress.${stemperature}K.${epercentage}e.out screen no

# Use cfg for AtomEye
dump 1 all cfg ${atomrate} dump._*.cfg id type xs ys zs c_myPE c_myKE c_myCN fx fy fz
dump_modify 1 element Cu

# Display thermo
variable thermostep equal "v_time_run/10"
thermo ${thermostep}
thermo_style custom "v_strain" pxx pyy pzz lx ly lz temp etotal pe ke
run ${time_run}

unfix def1
unfix 1
unfix 2

# SAVE THE DATA OF THE CALCULATION OR ELSE YOU NEED TO START OVER

write_restart restart.equil

# SIMULATION DONE
clear
print "creo ya esta =)"
```

The steps taken for the Uniaxial compression of copper were the following :

1. Equilibration of the system
2. Deformation of the computational box along the x-direction

3. Calculation of the stress-strain curve and shear-strain curve

4. Dump of the atomic trayectories for visualization

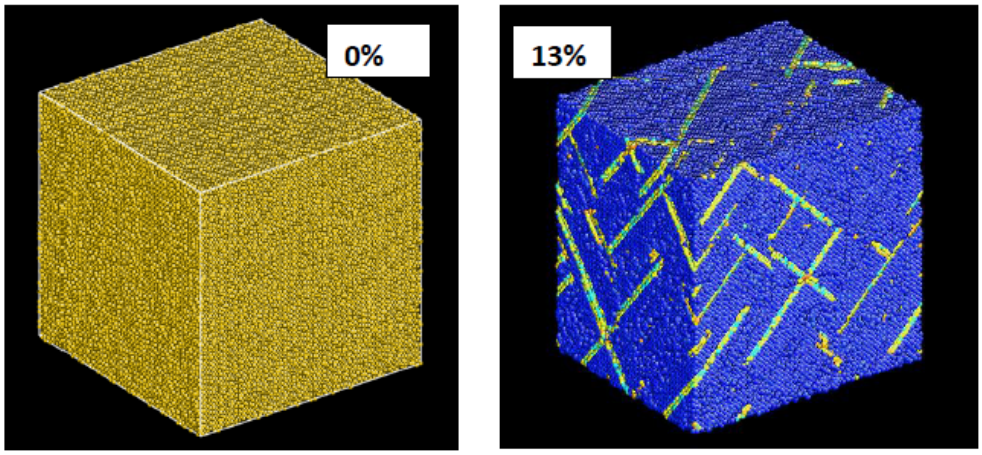

Figure 1.1: Uniaxial compression along (100) in Copper (500,000 atoms) at (a) 0% and (b) 13% of compression. The strain-rate used was 0.005 1/ps. The sudden decrease in the shear 0.5(pxx-0.5(pyy+pzz)) indicates where the elastic-plastic transition occurs. The visualization of the defects its done via a centro-symmetry parameter.

1.2 Quasi-Isentropic compression in Copper (SCRIPT2):

Quasi-isententropic compression refers to the process of compressing without a change in entropy. Quasi-isentropic compression is ideal for high strain deformations because there is not enough time for heat transfer.

From the first law of thermodynamics we can state that the heat flux is zero when the total Energy rate is equal to the mechanical work rate, in other words [2]:

$$\dot{E} = \dot{Q} + \dot{W} = \dot{\varepsilon}_\alpha V \sigma_{\alpha\alpha} = \sigma_{\alpha\alpha}\dot{V} = \dot{W} \rightarrow \dot{Q} = T\dot{S} = 0 \tag{1.1}$$

We implemented a new time profile deformation function in order to compress the material Quasi-isentropically. The new modified profile was added to fix_deform.cpp and fix_deform.h

The deformation of a body is given by $L(t) = Lo(1+\varepsilon)$, where Lo is the initial length of the computational box and (epsilon) is the strain . Here epsilon was defined as [6]:

$$\varepsilon = \frac{e_{max}\,(2t^3 - 3\tau t^2)}{6\tau^2} \rightarrow \dot{\varepsilon} = e_{max}\left(\frac{t}{\tau}\right)\left(1 - \frac{t}{\tau}\right) \tag{1.2}$$

Where emax is the maximum value located at $\tau/2$, therefore the simulation runs from t:{0,τ }. Its worth noticing that the only condition necessary to compress the material quasi-isentropic is given by the time profile deformation function i.e It can be any arbritary profile , but the condition is $\dot{\varepsilon}(t_o) = \dot{\varepsilon}(t_f)$ where t_o is the initial time and t_f is the final time.

The modification of LAMMPS source code to add the new deformation time profile is easy. In older versions of LAMMPS it was neccesary to HACK the code and modify fix_deform.cpp, but the recent version of LAMMPS include a new fix deform style "variable" : The variable style changes the specified box length dimension by evaluating a variable, which presumably is a function of time. I decided to implement my own HACK rather than use the fix deform "variable" style.

First I added a new variable (tau) to fix_deform.h , then we replaced the TRATE L(t) profile :

```
 set[i].lo_stop = 0.5*(set[i].lo_start+set[i].hi_start) -
 0.5*((set[i].hi_start-set[i].lo_start) *
 exp(set[i].rate*delt)); set[i].hi_stop = 0.5*(set[i].lo_start+set[i].hi_start) +
 0.5*((set[i].hi_start-set[i].lo_start) * exp(set[i].rate*delt));
```

to the new form given in equation (1.2) :

```
 set[i].lo_stop = 0.5*(set[i].lo_start+set[i].hi_start)
-0.5*((set[i].hi_start-set[i].lo_start) *
 (1 + pow(set[i].tau,-2)*0.1666*set[i].rate*(2*pow(delt,3)-3*set[i].tau*pow(delt,2))));
 set[i].hi_stop = 0.5*(set[i].lo_start+set[i].hi_start) +
 0.5*((set[i].hi_start-set[i].lo_start) *
 (1 + pow(set[i].tau,-2)*0.1666*set[i].rate*(2*pow(delt,3)-3*set[i].tau*pow(delt,2))));
```

Finaly I renamed TRATE L(t) profile to QRATE . After the modifications to fix_deform.cpp , and fix_deform.h then LAMMPS was recompiled e.g make openmpi. If LAMMPS compiles successfully, then the new profile is ready to RUN .

Now I will show a LAMMPS script that implements Quasi-isentropic compression in Copper:

```
# Quasi Isentropic deformation Copper along [100], [110] or [111]
# The equilibration step allows the lattice to expand to a temperature of 300 K with a
# pressure of 0 kbar at each simulation cell boundary. Then, the simulation cell is
# deformed in the x-direction at, while the lateral boundaries are controlled using
# the NVT equations of motion to mantain the lenght of the computational box constant.
# The stress and strain values are output to a separate file, which can be imported in
# a graphing application for plotting. The cfg dump files include the x, y, and z
# coordinates, the centrosymmetry values, the potential energies, and forces for each atom.
# This can be directly visualized using AtomEye We assume 1 timestep is equal to 0.0005
# pico-seconds

# SCRIPT MADE BY OSCAR GUERRERO-MIRAMONTES

# ---------- Initialize Simulation --------------------
clear
units metal
dimension 3
```

```
boundary p p p
atom_style atomic
atom_modify map array

# ---------- define variables  ---------------------

variable stemperature equal 300 # temperature in kelvin
variable epercentage equal 0.19 # the percentage the body is compressed
variable myseed equal 12345     # the value seed for the velocity
variable atomrate equal 250     # the rate in timestep that atoms are dump as CFG
variable time_step equal 0.0005 # time step in pico seconds (CHANGE THIS VALUE)
variable tau equal 5            # quasi-isentropic scheme final run time (in time UNITS)
variable time_eq equal 10000    # time steps for the equlibration part
variable time_nvt equal 5000    # time steps for the NVT (NUCLEATION DEFECTS)
variable tdamp equal "v_time_step*100"   # DO NOT CHANGE
variable pdamp equal "v_time_step*1000"  # DO NOT CHANGE
variable R equal "(v_epercentage/0.165)/v_tau" # Compress rate in EMAX
variable time_run equal "v_tau/v_time_step"    # time step for the deformation part

timestep ${time_step} # DO NOT CHANGE

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

  lattice    fcc 3.615 origin 0.0 0.0 0.0 orient x 1 0 0 orient y 0 1 0  orient z 0 0 1
# lattice    fcc 3.615 origin 0.0 0.0 0.0 orient x 1 1 0 orient y 0 0 1  orient z 1 -1 0
# lattice    fcc 3.615 origin 0.0 0.0 0.0 orient x 1 1 1 orient y 1 -1 0 orient z 1 1 -2

region box block 0 50 0 50 0 50 units lattice
create_box 1 box
create_atoms 1 box

# ---------- Define Interatomic Potential ---------------------

pair_style eam/alloy
pair_coeff * * Cu_mishin1.eam.alloy Cu

# ---------- Define Settings ---------------------------------

# "compute" Define a computation that will be performed on a group of atoms. Quantities
# calculated by a compute are instantaneous values, meaning they are calculated from
# information about atoms on the current timestep or iteration, though a compute may
# internally store some information about a previous state of the system. Defining a
# compute does not perform a computation.Instead computes are invoked by other LAMMPS
# commands as needed
```

```
  compute      myCN all cna/atom 3.8
  compute      myKE all ke/atom
  compute      myPE all pe/atom

# ---------- Equilibration ---------------------------------------

reset_timestep 0
velocity all create ${stemperature} ${myseed} mom yes rot no
fix equilibration all npt temp ${stemperature} ${stemperature} ${tdamp}
iso 0 0 ${pdamp} drag 0.0

# Output thermodynamics into outputfile
# for units metal, pressure is in [bars] = 100 [kPa] = 1/10000 [GPa]
# p2, p3, p4 are in GPa

variable eq1 equal "step"
variable eq2 equal "pxx/10000"
variable eq3 equal "pyy/10000"
variable eq4 equal "pzz/10000"
variable eq5 equal "lx"
variable eq6 equal "ly"
variable eq7 equal "lz"
variable eq8 equal "temp"
variable eq9 equal "etotal"
fix output1 all print 10 "${eq1} ${eq2} ${eq3} ${eq4} ${eq5} ${eq6} ${eq7} ${eq8} ${eq9}"
file run.${stemperature}K.out screen no

thermo 10
thermo_style custom step pxx pyy pzz lx ly lz temp etotal

# Use cfg for AtomEye (here only print the coordinates of the atoms)
dump 1 all cfg ${time_eq} fcc.copper.*.cfg id type xs ys zs
dump_modify 1 element Cu

# RUN AT LEAST 10000 timesteps
run ${time_eq}

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

#reset
unfix equilibration
undump 1
```

```
unfix output1

# ---------- Deformation ---------------------------------

reset_timestep 0

# QRATE IS OUR MODIFIED FUNCTIONAL FORM AND IS A LAMMPS HACK

fix 1 all nve
fix 2 all deform 1 x qrate ${R} ${tau} units box remap x

# Output strain and stress info to file
# for units metal, pressure is in [bars] = 100 [kPa] = 1/10000 [GPa]
# pxx, pyy, pzz are in GPa

variable strain equal "-(lx - v_L0)/v_L0"
variable shear  equal "0.5*(pxx/10000 - 0.5*(pyy/10000 + pzz/10000))"
variable tstep  equal "step"
variable out1 equal "v_strain"
variable out2 equal "pxx/10000"
variable out3 equal "pyy/10000"
variable out4 equal "pzz/10000"
variable out5 equal "v_shear"
variable out6 equal "lx"
variable out7 equal "ly"
variable out8 equal "lz"
variable out9 equal "temp"
variable out10 equal "vol"           # Volume Angstroms^3
variable out11 equal "etotal"        # ENERGY IN EV
variable out12 equal "press/10000"   # PRESSURE IN GPa
variable out13 equal "((press/10000)*vol)/160.21"  # MECHANICAL WORK W = PV in EV

fix def1 all print 1 "${tstep} ${out2} ${out3} ${out4} ${out5} ${out6} ${out7}
${out8} ${out9} ${out10} ${out11} ${out12}" file stress.${stemperature}K.${epercentage}e.out
screen no

# Display thermo
variable thermostep equal "v_time_run/10"
thermo ${thermostep}
thermo_style custom "v_strain" pxx pyy pzz lx ly lz temp etotal pe ke

# Use cfg for AtomEye (here only print the coordinates of the atoms)
dump 1 all cfg ${atomrate} dump.copper.*.cfg id type xs ys zs
dump_modify 1 element Cu
```

```
run ${time_run}

unfix def1
unfix 1
unfix 2

# SIMULATION DONE
clear
print "creo ya esta =)"
```

The above script first equlibrates the systems to a temperature of 300K , and next Quasi-isentropically compress the material up to 19% in 5 picoseconds. We can make a plot of the total energy rate $\dot{E}$ vs the mechanical work rate $\dot{W} = P\dot{V}$. As the next figure shows (figure 1.2), all the Energy transfer to mechanical work i.e $\dot{E} = \dot{W}$, thus the heat flux $\dot{Q}$ equals zero and the change of entropy with time is zero $\dot{S} = 0$

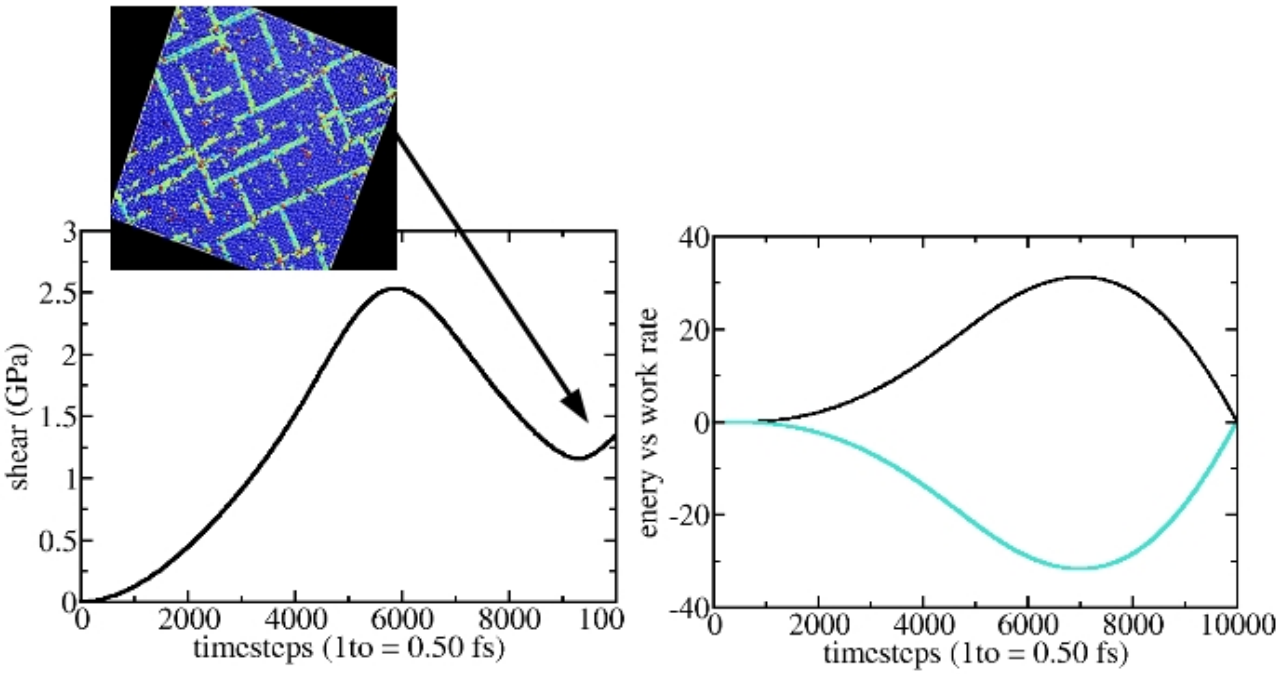

Figure 1.2: Uniaxial Quasi-isentropic compression along (100) in Copper (500,000 atoms). The image to the left show the Shear $\tau = 0.5(Pxx - 0.5(Pyy + Pzz))$ and the nucleation of defects using a centro-symmetry analysis. The right image show the energy rate $\dot{E}$ (black line) vs $\dot{W} = P\dot{V}$ (cyan line). As is evident $\dot{E} = \dot{W}$ and thus the $\dot{S} = 0$

Now I will show the time profile of pressure,temperature and strain rate obtained:

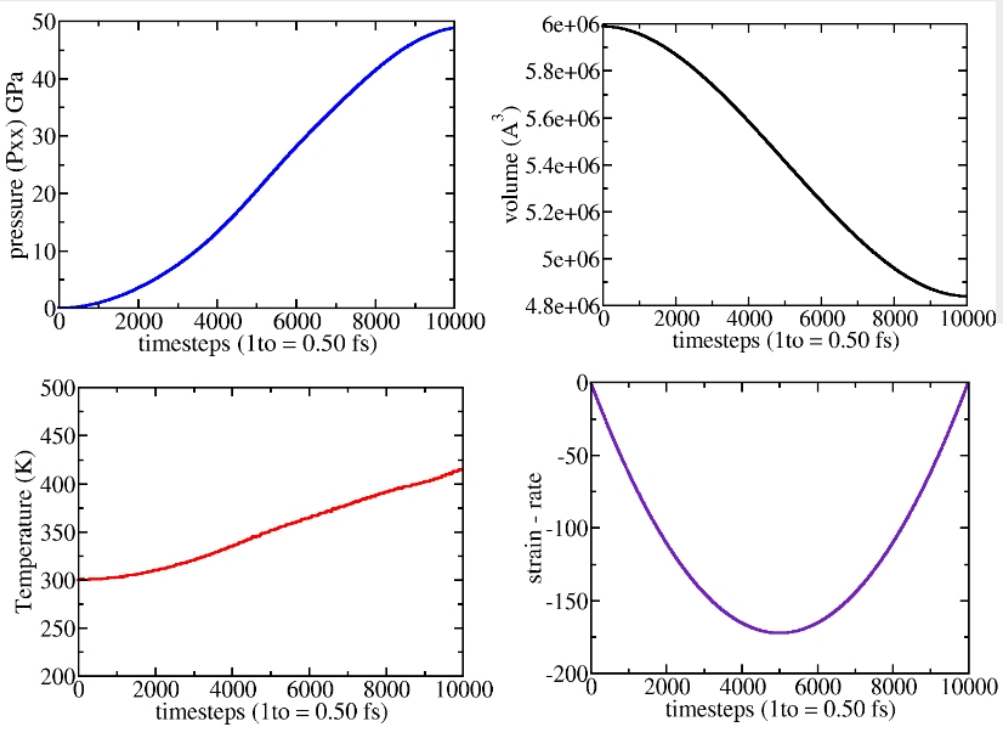

Figure 1.3: Uniaxial Quasi-isentropic compression along (100) in Copper (500,000 atoms). The image show the profiles of pressure, temperature, strain-rate and volume obtained after running the modified HACK version of LAMMPS

1.3 Shock Compression in Niquel (SCRIPT3):

Consider a continuous block of material of constant cross-section A. A piston drives into the material with a velocity U_p. In the situation being considered here, the piston is travelling at a constant velocity. Therefore the material behind the shock is travelling at the same velocity as the piston. From figure 1.4 and 1.5 it can be seen that the motion of the piston causes an increase in the density of the material ahead of it, as the material is 'gathered up'. The action of the piston, which causes the material to begin moving, and the resultant increase in density causes a shock to travel forward at velocity U_s.

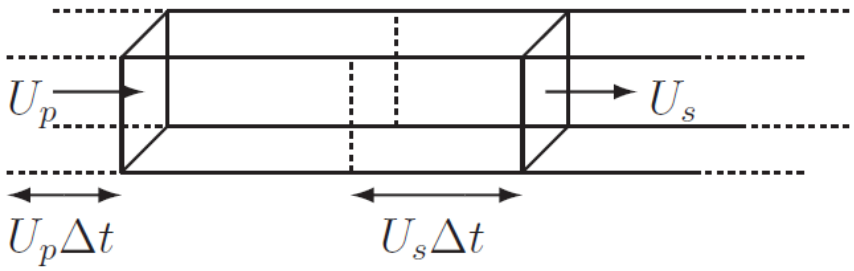

Figure 1.4: Schematic of shock propagation

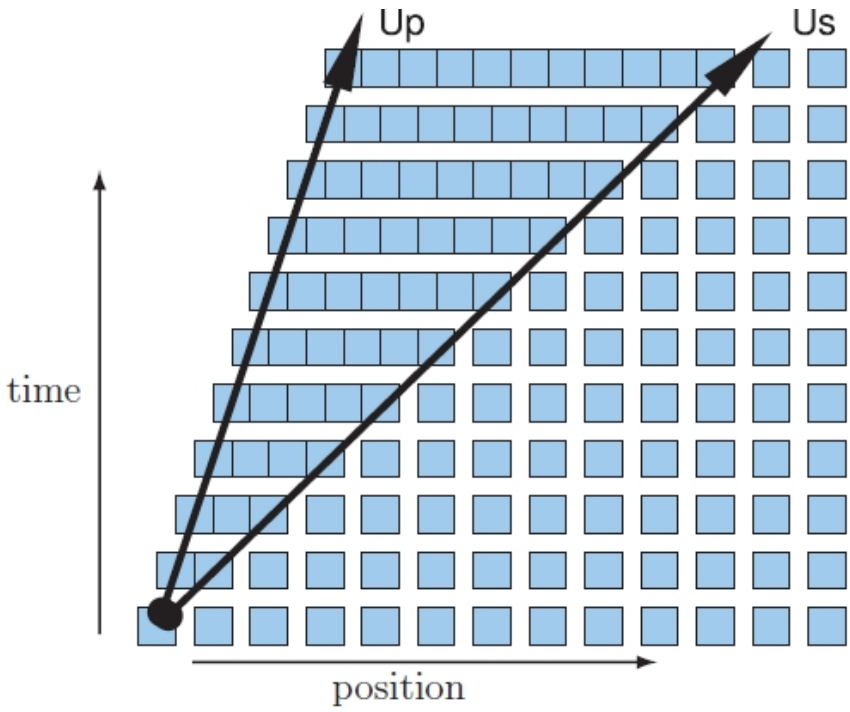

Figure 1.5: Schematic of a simple piston driven shock. The piston drives into the material from the left. Each horizontal row is a snapshot of the system at some time. The initial state is at the bottom of the figure, and time increases up the figure. Thus it can be seen that the slope of the arrow labelled Up gives the velocity of the piston, and that labelled Us gives the velocity of the shock front running ahead into the material at rest.

Now I will present a LAMMPS's Script that simulates a piston drive shock as described in the above figure. In this example we choose FCC Niquel [7,8]. Descriptions of the physics of shock and calculation of the shock speed U_s are given in many texts. The interested reader is referred to [9,10,11,12,13,14].

```
# SCRIPT MADE BY: OSCAR GUERRERO

# ---------- Initialize Simulation --------------------

clear
units metal
dimension 3
boundary p p p
atom_style atomic
atom_modify map array

# ---------- define variables  --------------------

variable stemperature equal 5     # temperature in kelvin
variable alattice     equal 3.520 # lattice constant (unit A)
variable myseed       equal 12345 # the value seed for the velocity
variable xmax         equal 40    # size in the x-direction
variable ymax         equal 40    # size in the y-direction
```

```
variable zmax         equal 70    # size in the z-direction
variable time_step    equal 0.001 # time step in pico seconds
variable time_eq      equal 1000  # time steps for the equlibration part
variable time_shock   equal 15000 # time steps for the piston
variable vpiston      equal 1.300 # piston speed in (km/s) multiply by ten to obtain (A/ps)
variable Nevery       equal 10    # use input values every this many timesteps
variable Nrepeat      equal 5     # number of times to use input values for calculating
variable Nfreq        equal 100   # calculate averages every this many timesteps
variable deltaz       equal 3     # thickness of spatial bins in dim (distance units)
variable atomrate     equal 100   # the rate in timestep that atoms are dump as CFG
variable tdamp equal  "v_time_step*100"  # DO NOT CHANGE
variable pdamp equal  "v_time_step*1000" # DO NOT CHANGE

# DO NOT CHANGE
variable Up equal "10*v_vpiston"

timestep ${time_step}

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

  lattice fcc ${alattice} origin 0.0 0.0 0.0 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1
# lattice fcc ${alattice} origin 0.0 0.0 0.0 orient x 1 1 0 orient y 0 0 1 orient z 1 -1 0
# lattice fcc ${alattice} origin 0.0 0.0 0.0 orient x 1 1 1 orient y 1 -1 0 orient z 1 1 -2

# the create box commands create_box natoms idname, where natoms is the number of atoms type
# in the simulation box for this case we only have one type of atoms
# you can use the command "set" to set the different region of atoms to different Ntype
# the command "set" is very useful examples: "set group piston type 1"

     # define size of the simulation box
     region sim_box block 0 ${xmax} 0 ${ymax} 0 ${zmax} units lattice
     create_box 2 sim_box

     # define atoms in a small region
     region atom_box block 0 ${xmax} 0 ${ymax} 0 ${zmax}
     create_atoms 1 region atom_box

     # define a group for the atom_box region
     group atom_box region atom_box

region piston   block INF INF INF INF INF 4
region bulk     block INF INF INF INF 4 INF

group piston    region piston
group bulk      region bulk
```

```
set  group piston type 1
set  group bulk   type 2


# ---------- Define Interatomic Potential ---------------------

pair_style eam/fs
pair_coeff * * Ni.lammps.eam Ni Ni

       mass 1 58.70
       mass 2 58.70

 compute      myCN   bulk cna/atom fcc
 compute      myKE   bulk ke/atom
 compute      myPE   bulk pe/atom
 compute      myCOM  bulk com
 compute    peratom  bulk stress/atom
 compute      vz     bulk property/atom vz

# ------------ Equilibrate ----------------------------------------

reset_timestep 0

# Now, assign the initial velocities using Maxwell-Boltzmann distribution

velocity atom_box create ${stemperature} ${myseed} rot yes dist gaussian
fix equilibration bulk npt temp ${stemperature} ${stemperature} ${tdamp}
 iso 0 0 ${pdamp} drag 1

variable eq1 equal "step"
variable eq2 equal "pxx/10000"
variable eq3 equal "pyy/10000"
variable eq4 equal "pzz/10000"
variable eq5 equal "lx"
variable eq6 equal "ly"
variable eq7 equal "lz"
variable eq8 equal "temp"
variable eq9 equal "etotal"
fix output1 bulk print 10 "${eq1} ${eq2} ${eq3} ${eq4} ${eq5} ${eq6} ${eq7} ${eq8} ${eq9}"
 file run.out screen no

thermo 10
thermo_style custom step pxx pyy pzz lx ly lz temp etotal
```

```
run ${time_eq}

unfix equilibration
unfix output1

# -------------- Shock  ------------------------------------------

change_box all boundary p p s
reset_timestep 0

# WE CREATE A PISTON USING A FEW LAYERS OF ATOMS AND THEN WE GIVE IT
# A CONSTANT POSTIVE SPEED. YOU COULD ALSO USE LAMMPS' FIX  WALL/PISTON COMMAND

fix 1 all nve
velocity piston set 0 0 v_Up sum no units box
fix 2 piston setforce 0.0 0.0 0.0

# WE CREATE BINS IN ORDER TO TRACK THE PASSING OF THE SHOCKWAVE

fix density_profile bulk ave/spatial ${Nevery} ${Nrepeat} ${Nfreq} z lower ${deltaz}
 density/mass file denz.profile units box

variable temp atom  c_myKE/(1.5*8.61e-5)
fix temp_profile bulk ave/spatial ${Nevery} ${Nrepeat} ${Nfreq} z lower ${deltaz}
 v_temp file temp.profile units box

variable meanpress atom -(c_peratom[1]+c_peratom[2]+c_peratom[3])/3
fix pressure_profile bulk ave/spatial ${Nevery} ${Nrepeat} ${Nfreq} z lower ${deltaz}
 v_meanpress units box file pressure.profile

fix velZ_profile bulk ave/spatial ${Nevery} ${Nrepeat} ${Nfreq} z lower ${deltaz}
 c_vz  units box file velocityZcomp.profile

variable eq1 equal "step"
variable eq2 equal "pxx/10000"
variable eq3 equal "pyy/10000"
variable eq4 equal "pzz/10000"
variable eq5 equal "lx"
variable eq6 equal "ly"
variable eq7 equal "lz"
variable eq8 equal "temp"
variable eq9 equal "etotal"
variable eq10 equal "c_myCOM[3]"
fix shock bulk print 10 "${eq1} ${eq2} ${eq3} ${eq4} ${eq5} ${eq6} ${eq7} ${eq8} ${eq9}
${eq10}" file run.${stemperature}K.out screen no
```

```
thermo 10
thermo_style custom step pxx pyy pzz lx ly lz temp etotal c_myCOM[3]

#Use cfg for AtomEye
  dump 1 all cfg ${atomrate} dump._*.cfg id type xs ys zs c_myPE c_myKE c_myCN
  dump_modify 1 element Cu Ni

run ${time_shock}
```

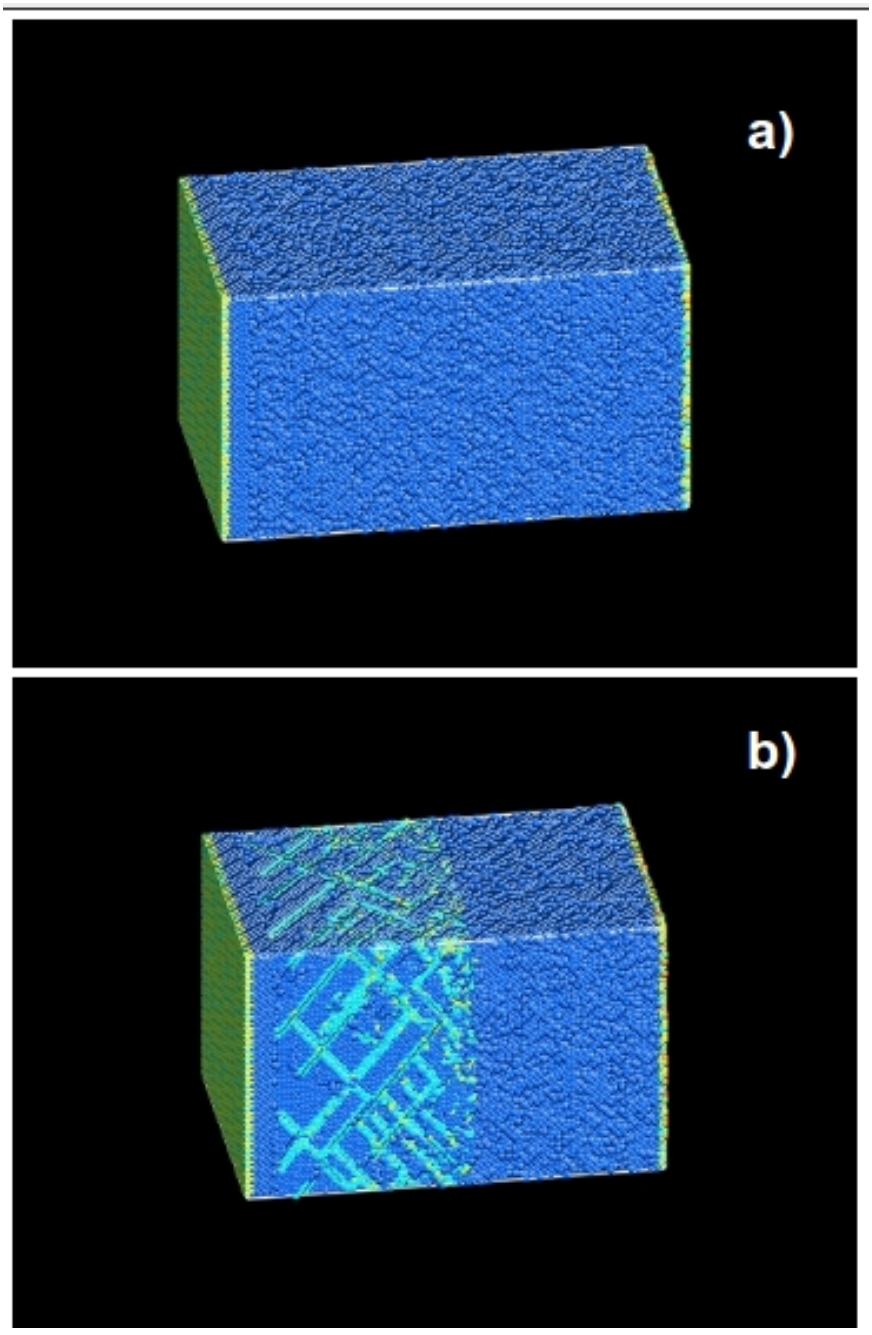

Figure 1.6: NEMD shock compression simulation in Niquel using the EAM potential interaction and a piston speed Up = 1.3 km/s. Figure a). shows the system at the beggining and figure b). Shows the nucleation of defects due to the passing of the shockwave.

The steps taken for the Shock compression of Niquel were the following :

1. Equilibration of the system
2. Define a fixed layer of atoms to act as a Piston
3. Periodic boundary conditions are applied in the x and y directions. Boundaries are free in the z direction
4. Define BINS to track the passing of the shockwave
5. Dump of the atomic trayectories for visualization

Acknowledgement

I would like to thank the LAMMPS community for the great FeedBack and discussions that made it possible to complete this brief tutorial. I would also like to thank the following people for their contribution of ideas: Steve Plimpton, Nigel Park, Ray Shan, Jonathan Zimmerman, Vishnu Wakof, Anirban Dhar, Axel Kohlmeyer. Selesta Oxem, Rajdeep Behera, Paul Swain, Gildardo Rivas, and Mark Tschopp

REFERENCES

[1] S.J. Plimpton, Fast parallel algorithms for short-range molecular dynamics, J Comp Phys 117 (1995) 1.

[2] O. Guerrero, Pressure induced dynamical instabilities in body center cubic crystals, Thesis, M.S., UTEP, 2010, p. 35.

[3] Yuhang Jing; Qingyuan Meng; Yufei Gao, Molecular dynamics simulation on the buckling behavior of silicon nanowires under uniaxial compression, Computational Materials Science (April 2009), 45 (2), pg. 321-326

[4] J. Li, AtomEye: an efficient atomistic configuration viewer, Modelling Simul. Mater. Sci. Eng. 11 (2003) 173.

[5] Y. Mishin, M. J. Mehl, D. A. Papaconstantopoulos, A. F. Voter and J. D. Kress, Structural stability and lattice defects in copper: Ab initio, tight-binding, and embedded-atom calculations, Phys. Rev. B 63, 224106 (2001)

[6] R. Ravelo, B.L. Holian, c T.C. Germann, High strain rates effects in quasi-isentropic compression of solids, AIP Conf. Proc. 1195 (2009).

[7] H N Jarmakani, E M Bringa, P Erhart, B A Remington, Y M Wang, N Q Vo, M A Meyers, Molecular dynamics simulations of shock compression of nickel: From monocrystals to nanocrystals, Acta Materialia 56 (2008) 5584-5604

[8] Y. Mishin, D. Farkas, M.J. Mehl, and D.A. Papaconstantopolous, Interatomic Potentials for Monoatomic Metals from Experimental Data and ab initio calculations, Phys. Rev. B 59, 3393 (1999)

[9] E.M. Bringa, J.U. Cazamias, P. Erhart, J. Stolken, N. Tanushev, B.D. Wirth, et al., Atomistic shock Hugoniot simulation of single-crystal copper, J. Appl. Phys. 96 (2004) 3793.

[10] Brad Lee Holian, Plasticity Induced by Shock Waves in Nonequilibrium Molecular-Dynamics Simulations, Proceedings of APS Topical Group on Shock Compression, Amherst, MA, 27 July - August (1997)

[11] Lan He, Thomas D. Sewell, and Donald L. Thompson, Molecular dynamics simulations of shock waves in oriented nitromethane single crystals, J. Chem. Phys. 134, 124506 (2011)

[12] W. J. Macquorn Rankine. On the thermodynamic theory of waves of finite longitudinal disturbance. Philosophical Transactions of the Royal Society of London, 160:277-288, January 1870.

[13] A. Siavosh-Haghighi, R. Dawes, T. D. Sewell, and D. L. Thompson, Shock-induced melting of (100)-oriented nitromethane: Energy partitioning and vibrational mode heating, J. Chem. Phys. 131, 064503 (2009).

[14] H. Hugoniot. Propagation des mouvements dans les corps et specialement dans les gaz parfaits. Journal de lEcole Polytechnique, 57:3-98, 1887.