

Foundations of Computational Materials Modelling

Narasimhan Swaminathan

Department of Mechanical Engineering

Indian Institute of Technology Madras

LAMMPS exercises 5

(Refer Slide Time: 0:24)

Elastic constants of a cubic material

Simple method to determine elastic constants using `fix deform` command. We use `Al_jnp.eam` as we used in exercise13. Since this is cubic a material we will have C_{11} , C_{12} and C_{44} . These are C_{1111} , C_{1122} and C_{2323} . We can use the relation between the stress and strain, to determine the constants. Consider the constitutive relation (in Voigt notation)

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix} \quad (45)$$

Narasimhan Swaminathan (IITM) ME7244 October 29, 2019 45 / 47

Introduction to LAMMPS

- VIC 12-Jun-2019 at 7:44 AM 130.4 KB Application
- VMware Workstation Client 30-Aug-2018 at 12:32 AM 81.4 KB Application
- VMware Horizon Client 01-Mar-2019 at 6:48 PM 114.8 KB Application
- Word Menus 20-Aug-2019 at 1:07 PM 2.1 KB Application
- Wacom Tablet 24-Oct-2019 at 4:48 PM - Folder
- lammmps.pdf 08-Oct-2019 at 12:58 PM 16.5 KB Application



Elastic con

```

Al_jnp.eam      deformensile.lammpsmp
C11AL.m        log.lammps
Narasimhanhomeair:exercise15 narasimhan$ cd ../
Narasimhanhomeair:Argon narasimhan$ ls
Ar_1000_Kr_0.txt      exercise14
Argon_Example_Inputs exercise15
Argon_Example_Inputs.zip exercise2
DeepakB.1            exercise3
Exercise_list.doc     exercise4
Exercise_list.odt     exercise5
FCC_Argon.in         exercise6
Icon?                 exercise7
Important_Argon_Paper_2017.pdf exercise8
Si.sw                 exercise8.1
dump.eqm1             exercise9
dump.eqm2             exerciserdf
dump.eqm3             exerciserdfgas
exercise1             in.threbody
exercise10            input_argon.in
exercise11            log.lammps
exercise12            log_n_of_1000.txt
exercise13            logfile.txt
exercise13            output.out
Narasimhanhomeair:Argon narasimhan$ cd exercise10/
  
```

Narasimhan Swaminathan (IITM) ME7244 October 29, 2019 45 / 47

Introduction to LAMMPS

- VIC 12-Jun-2019 at 7:44 AM 130.4 KB Application
- VMware Workstation Client 30-Aug-2018 at 12:32 AM 81.4 KB Application
- VMware Horizon Client 01-Mar-2019 at 6:48 PM 114.8 KB Application
- Word Menus 20-Aug-2019 at 1:07 PM 2.1 KB Application
- Wacom Tablet 24-Oct-2019 at 4:48 PM - Folder
- lammmps.pdf 08-Oct-2019 at 12:58 PM 16.5 KB Application



```

15 #-----
16 mass 1 39.948 #Mass of atom type 1 is 39.48 [mass units grams/mole]
17 pair_style lj/cut 20.0 # k_B = 8.6173303e-5 eV/K #<How are atoms interacting. Provide the name of the p
omential and the corresponding cut-off distance>
18 pair_coeff 1 1 0.01006418 3.3952 #<The coefficient of the lj potential for the interactions of at
om type 1 with 1>
19 #-----
20 variable v1 equal vol
21 print "Before minimization $(v1)"
22 group ar type 1 #<Group all the argon types (argon type is of type 1. All atoms of type 1 are in group ar)
23 group atom2 id 2
24 fix relaxBox all box/relax iso 1.0
25 minimize 1e-10 1e-9 100000 100000 #<Minimize the energy using a conjugate gradient step.
26 #-----
27 variable v2 equal vol
28 variable v2 equal vol
29 print "After minimization $(v2)"
30 #Provide an initial Maxwellian distribution of velocity corresponding to temperature 5K
31 velocity all create 600 198728 dist gaussian
32 #Perform an NVE integration with this initial position and velocity distribution
33 timestep 0.001 #<Time step in ps. So this is 0.001ps or 1 femto second>
34 fix 1 all nve
35 thermo_style custom step time cpu temp pe ke etotal press vol#<Print the thermodynamic information >
36 thermo 1000
37 run 10000
38 unfix 1
39 fix 1 all npt temp 300 300 0.1 iso 1.0 1.0 1.0
40 run 50000
41 unfix 1
42 #-----
43 #Check for equilibrium
44 fix 1 all nve
45 run 10000
46 unfix 1
47 #-----
48 #
49 log rdflog.txt
50 reset timestep 0
51 compute rdfs all rdf 100 1 1 cutoff 20.0
52 fix rdf all ave/time 1 10000 10000 c_rdfs[1] c_rdfs[2] c_rdfs[3] file rdf_gas.txt mode vector
53 thermo 1
54 fix 1 all nve
55 run 20000

```



```

AL.in C13AL.m deformtensile.lampstrj strain.txt
Al_jnp.eam DELTZ.txt log.lammps
Narasimhanhomeair:exercise15 narasimhan$ vim AL.in
Narasimhanhomeair:exercise15 narasimhan$ vim strain.txt
Narasimhanhomeair:exercise15 narasimhan$ ls
AL.in C13AL.m deformtensile.lampstrj strain.txt
Al_jnp.eam DELTZ.txt log.lammps
Narasimhanhomeair:exercise15 narasimhan$ ls
AL.in C13AL.m deformtensile.lampstrj strain.txt
Al_jnp.eam DELTZ.txt log.lammps
Narasimhanhomeair:exercise15 narasimhan$ vim AL.in
Narasimhanhomeair:exercise15 narasimhan$ ls
AL.in C13AL.m deformtensile.lampstrj strain.txt
Al_jnp.eam deformtensile.lampstrj
C13AL.m log.lammps
Narasimhanhomeair:exercise15 narasimhan$ cd ../
Narasimhanhomeair:Argon narasimhan$ ls
Ar_1000_ar_d.txt exercise14
Argon_Example_Inputs exercise15
Argon_Example_Inputs.zip exercise2
Dewask8.1 exercise3
Exercise_list.doc exercise4
Exercise_list.out exercise5
FCC_Argon.in exercise6
Icon? exercise7
Important_Argon_Paper_2017.pdf exercise8
SI.sm exercise8.1
dump.eqm1 exercise9
dump.eqm2 exerciserd
dump.eqm3 exerciserdgas
dump.min in.threebody
exercise1 input_argon.in
exercise10 log.lammps
exercise11 log-n_ar_1000.txt
exercise12 logfile.txt
exercise13 output.out
Narasimhanhomeair:Argon narasimhan$ cd exercise10/
Narasimhanhomeair:exercise10 narasimhan$ ls
RDF_gas.aux RDF_gas.tex rdf_gas.txt
RDF_gas.log Testifave.xlsx rdflog.txt
RDF_gas.pdf input10_argon.in
RDF_gas.synctex.gz log.lammps
Narasimhanhomeair:exercise10 narasimhan$ vim input10_argon.in
Narasimhanhomeair:exercise10 narasimhan$ vim r

```



So, good afternoon. Let us continue with where we left from, where we left off. So, basically in today's class what I wanted to tell you was I wanted to clarify some of the issues concerning this the frequency with which you have to n every, n repeat, that particular issue that we had yesterday. So, I was hoping that you would have gone back and tried out what happens when you, so that was exercise number 10. So, when I basically write at the end when I used reset underscore time step 0 and then if the same understanding that we had previously, if we calculate these average time values then this one has to be the frequency and this one has to be how, this is n every.

So, every so many inputs has to be taken amongst the 10000 different configurations and we take all the 10000 configurations in order to perform the average. So, when I did this and I looked at my rdf underscore gas dot txt, I found two sets of information like what I should

want for 20000. The next part 20000 does not basically contain the header, so you might get confused that it is not printing out the next set of information. So, it works out really fine. And I also the best thing to do if at all you have such doubts is to run it for a small period of time and manually average it to see actually see whether it is doing what you want, that is what I did for this and it seems to be working perfectly fine. So, our understanding concerning what these three, value these three values are was actually correct. So, I hope you had an opportunity to try that yesterday.

So, today what we will do is we will we saw a couple of different potentials, we can keep talking about different kinds of potentials and how to give, how to feed them into LAMMPS and how to make the input script read the information, but I thought today I will actually provide you and provide you with an input script that will essentially calculate the elastic constants of a solid. So, what we are going to do is we are going to use the same idea, same interatomic potential that we used for aluminium.

And we are going to calculate its elastic constants, and all obviously, we need to be able to check whether the calculated values are close enough or not.

(Refer Slide Time: 2:54)

Aluminum Potentials	C ₁₁	Error	C ₁₂	Error	C ₄₄	Error
Mishin Ni Al Cu 2013.eam.alloy ⁴¹	107.21	0.08%	60.60	0.37%	32.88	16.18%
ADP9.eam.alloy ⁴⁴	107.03	0.25	61.06	0.43	31.05	9.72
Mishin NIAI2009 ⁴²	107.5	0.19	61.25	0.74	33.2	17.31
Mishin Al Co 2013.eam.alloy ⁴¹	107.58	0.26	61.82	1.68	32.88	16.18
Mg Al set.eam.alloy ⁴⁶	107.08	0.21	58.35	4.03	31.6	11.66
Al Mg.eam.fs ⁴⁸	103.86	3.21	63.8	4.93	30.48	7.70
AlPb-seff.eam.alloy ⁴⁷	105.63	1.56	58.2	4.28	31.85	12.54
Zope Ti Al 2003.eam.alloy ⁴⁸	98.27	8.42	56.23	7.52	26.02	8.06
AP3.eam.alloy ⁴⁸	98.2	8.48	55.88	8.09	26.42	6.64
NiAl.eam.alloy ⁴⁹	100.52	6.32	54.65	10.12	26.14	7.63
AlFe.mn ⁵⁴	95.25	11.23	55.82	8.19	30.28	7.00
Al Fe.eam.fs ⁷⁹	94.15	12.26	55.18	9.24	30.28	7.00
Almn ²⁰	103.95	3.12	63.96	5.20	36	29.6%
Al.eam.fs ⁴²	101.53	5.38	64.5	6.09	36	29.6%
Al jip.eam ⁸⁰	96.58	9.99	74.61	22.71	30.03	42.56
Al LEA.eam.alloy ⁷³	123.09	14.72	64.18	5.56	39.5	39.58
AlO.eam.alloy ⁷¹	82.92	22.72	69.3	13.98	30.37	7.31
Al zhuo.eam.alloy ⁶⁸	81.38	24.16	59.12	4.41	29.15	28.80
CoAl.eam.alloy ⁷²	83.16	22.50	55.1	9.37	21.41	24.35
AlCu.eam.alloy ³⁴	73.97	31.06	59.24	2.57	51.1	80.57
Farkas Nb Ti Al 1996.eam.alloy ⁷³	171.64	59.96	116.1	93.24	91.83	224.49
Alser ⁴²	61.16	43.00	45.74	24.77	10.56	61.63
NiAlH jca.eam.fs ⁷⁴	58.26	45.70	33.5	44.90	58.26	105.87
NiAlH jca.eam.alloy ⁷⁴	58.98	45.03	32.9	45.89	58.98	108.41



So, I am going to do some sort of a brute force method in order to calculate these elastic constants today, but it is but the methodology that we are following is quite intuitive. So, you will you will find it easy to understand. So, this will be exercise 14 and I have a paper here, where I have some document here which was got from the internet. So, I will share this with you as well and they have actually calculated the elastic constants of several metals and

alloys using embedded atom method and they have tabulated the values, so we will actually be using exactly this, if you remember right the file name that we used for the aluminium interatomic potential was also aluminium underscore jnp dot eam. And these are the values of the elastic constants that we need to calculate for this particular material.

(Refer Slide Time: 3:55)

Introduction to LAMMPS

Elastic constants of Al -exercise14

Elastic constants of a cubic material

Simple method to determine elastic constants using **fix deform** command. We use AL.jnp.eam as we used in exercise13. Since this is cubic a material we will have C_{11} , C_{12} and C_{44} . These are C_{1111} , C_{1122} and C_{2323} . We can use the relation between the stress and strain, to determine the constants. Consider the constitutive relation (in Voigt notation)

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix} \quad (45)$$

Narasimhan Swaminathan (IITM) ME7244 October 29, 2019 45 / 47

So, so, aluminium, as you might know is actually a cubic material. So, aluminium is actually a cubic metal. So, cubic metal means it will have 3 independent elastic constants basically C_{11} , C_{12} and C_{44} . So, C_{11} , C_{12} and C_{44} are basically the elastic constants written in the voigt notation. So, I hope you know what voigt notation is, this is essentially C_{1111} , C_{1122} and C_{2323} if you will. So, but for the cubic material, the relationship between the stress and the strain tensor can be written as equation number 45.

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix} \quad (45)$$

So, you see that C11, C22 and C33 are equal C12 and C13 are all, all these things are equal. So, this, this set, this set and then all these shared components all turn out to be equal for cubic material. So, using this constitutive relationship, we can design experiments using molecular dynamic simulations in order to determine these elastic constants.

(Refer Slide Time: 5:05)

Introduction to LAMMPS

Elastic constants

So we have first,

$$\sigma_{33} = C_{12}\epsilon_1 + C_{12}\epsilon_2 + C_{11}\epsilon_3 \quad (46)$$

If we keep strain $\epsilon_1 = \epsilon_2 = 0$ and apply only $\epsilon_3 = t\epsilon_3$, we have

$$C_{11} = \frac{\sigma_{33}}{\epsilon_3} \quad (47)$$


Similarly, with the same experiment we also have

$$\sigma_{22} = C_{12}\epsilon_1 + C_{11}\epsilon_2 + C_{12}\epsilon_3 \quad (48)$$

Since $\epsilon_1 = \epsilon_2 = 0$, we have

$$C_{12} = \frac{\sigma_{22}}{\epsilon_3} \quad (49)$$

Narasimhan Swaminathan (IITM) MET244 October 29, 2019 46 / 47



Introduction to LAMMPS


Elastic constants of Al -exercise14

Elastic constants of a cubic material

Simple method to determine elastic constants using **fix deform** command. We use Al.jnp.eam as we used in exercise13. Since this is cubic a material we will have C_{11}, C_{12} and C_{44} . These are C_{1111}, C_{1122} and C_{2323} . We can use the relation between the stress and strain, to determine the constants. Consider the constitutive relation (in Voigt notation)

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix} \quad (45)$$

Narasimhan Swaminathan (IITM) MET244 October 29, 2019 45 / 47



So, if you take a look at the first equation that we can write, so what what we can write for example is σ_{33} or σ_{33} is nothing but C_{12} times ϵ_1 , C_{12} times ϵ_2 and C_{11} times ϵ_3 . Suppose, we keep the strains ϵ_1 and ϵ_2 to be 0. So, we constrain the box to not be expanding or contracting in that direction and apply only a strain in the z direction at a specific rate, then C_{11} is nothing but σ_{33} divided by ϵ_3 provided the stress strain relationship is linear for the amount of strain that we have actually given.

$$\sigma_{33} = C_{12}\epsilon_1 + C_{12}\epsilon_2 + C_{11}\epsilon_3$$

$$C_{11} = \frac{\sigma_{33}}{\epsilon_3}$$

So, we need to make sure that we are going to deform this simulation cell very little, not too much. If you do too much then you will have yielding and, and all that so that you can we can see all that as well. Now the same experiment, it is possible for us to say that σ_{22} , with the same experiment, with basically ϵ_1 and ϵ_2 constraint with only ϵ_3 applied with the same experiment, you can also see that σ_{22} is actually equal to C_{12} times ϵ_1 , C_{11} times ϵ_2 , and C_{12} times ϵ_3 . So, since ϵ_1 and ϵ_2 are 0, C_{12} turns out to be σ_{22} by ϵ_3 . The slope of σ_{22} versus ϵ_3 will give you basically C_{12} .

$$C_{12} = \frac{\sigma_{22}}{\epsilon_3}$$

(Refer Slide Time: 6:45)


Introduction to LAMMPS

Elastic constants

The other constant C_{44} can be determined by applying a shear strain. We have

$$C_{44} = \frac{\sigma_6}{\epsilon_6} \quad (50)$$

Narasimhan Swaminathan (IITM) ME7244 October 29, 2019 47 / 47




Introduction to LAMMPS

Elastic constants of Al -exercise14

Elastic constants of a cubic material
 Simple method to determine elastic constants using **fix deform** command. We use Al.jnp.eam as we used in exercise13. Since this is cubic a material we will have C_{11} , C_{12} and C_{44} . These are C_{1111} , C_{1122} and C_{2323} . We can use the relation between the stress and strain, to determine the constants. Consider the constitutive relation (in Voigt notation)

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix} \quad (45)$$

Narasimhan Swaminathan (IITM) ME7244 October 29, 2019 45 / 47



However, for the other constant C_{44} , you need to apply a shear strength. So, σ_4 is equal to C_{44} times ϵ_4 or σ_5 is equal to, σ_5 means C_{44} times ϵ_5 , σ_6 is C_{44} times ϵ_6 , and the σ_{12} is basically C_{44} times ϵ_{12} or ϵ_{xy} . So,

we can do these simple experiments in LAMMPS in order to find out these elastic constants and which is what we will be doing in exercise number 14.

$$C_{44} = \frac{\sigma_6}{\epsilon_6}$$

(Refer Slide Time: 7:21)

The image shows a person sitting at a desk with a laptop. Overlaid on the scene is a terminal window displaying a file listing for exercise 14. The listing includes files such as exercise5 through exercise14, various input and output files, and a directory structure for exercise14. The terminal prompt is 'Narasimhanhomeair:exercise14 narasimhan\$ vim'.

```

1 units      metal
2 atom_style atomic
3 boundary   p p p
4
5 #fcc unit cell
6 variable a index 3.986
7 lattice fcc $a)
8 region     myreg block    0 10 &
9           0 10 &
10          0 20
11 #Lattice units is used, which is the default
12 #in the argon example, we used box units.
13 create_box 1 myreg
14 create_atoms 1 region myreg
15 # Minimize using Embedded atom method potential for Al
16
17 pair_style eam
18 pair_coeff * * Al_jnp.eam
19 minimize 1e-25 1e-25 100000 100000
20 #=====
21 #Total time is 10ps
22 #Total strain is 0.02
23 #Strain rate is 0.02 per ps, which is 0.002x10(12) strains per second
24 #2x10-9 strains per second!! Very high
25 #=====
26 timestep 0.001
27 reset_timestep 0
28 compute str all pressure NULL virial
29 variable l1 equal lz
30 variable ezz equal (l1-$l1)/$l1)
31 #&& defz all deform l z delta 0.0 10000 units box
32 dump dumpid all custom 100 deformtensile.lampstrj id x y z
33 thermo_style custom step time c_str[1] c_str[2] c_str[3] c_str[4] c_str[5] c_str[6] v_ezz lx ly
34 thermo 100
35 run 10000
36

```



So, σ_{xy} , so x is one direction and y is two direction and z is three direction. So, you are asking me, why is a σ_{11} and σ_{12} ? Is that your question or?

Student: I am just simply asking you meaning of 1 2 3?

Professor: x y and z directions. And we have used the white notation that to voigt the equation, you know the voigt notation?

Student: Yes, sir. I am aware of it, but I do not remember it at this time.

Professor: That is fine, that is fine. So basically, it just helps you to represent a 2nd order tensor as a vector. The stress and the strain and then you are able to represent the 4th order tensor as a 2nd order as a 2 by 2 matrix, sorry, as a 6 by 6 matrix. You can you can refer to any standard books on elasticity, you will find what exactly that means, it is very simple. Now, I want to focus on writing these input scripts.

So, in this case, I am going to be deforming the simulation box in the z direction. So as before, it is a same aluminium sample with Lattice constant being 3.986, which is exactly the value that gave us the minimum energy. I have just started off with that, I just started off with that configuration, defined a region where the z direction has 20 unit cells, whereas the x and the y direction have 10 unit cells. Created in the region, created the atoms inside that region, I am saying what interatomic potential that I am going to use and then defining the file that will contain the interatomic of potential. And then I quickly minimize it, this this step might not be required in this case.

But what we want to do is we want to basically move the simulation box. So now, you imagine a simulation box, which is 10 unit cells in the x direction, 10 unit cells in the y direction and 20 units is z direction. What we want to do is I want to move the top face that is the z top face by a certain amount, by a certain amount. The command in LAMMPS which allows me to do that is basically the fix deform command. So, fix deform command is what we will be using. So first what I do is I want to compute the stress. So, I say compute stress, all pressure null virial.

(Refer Slide Time: 10:04)

compute pressure command

Syntax

```
compute ID group-ID pressure temp-ID keyword ...
```

- ID, group-ID are documented in `compute` command
- pressure = style name of this compute command
- temp-ID = ID of compute that calculates temperature, can be NULL if not needed
- zero or more keywords may be appended
- keyword = ke or pair or bond or angle or dihedral or improper or kspace or fix or virial or pair/hybrid

Examples

```
compute 1 all pressure thermo_temp
compute 1 all pressure NULL pair bond
compute 1 all pressure NULL pair/hybrid 13/cut
```

Description

Define a computation that calculates the pressure of the entire system of atoms. The specified group must be "all". See the `compute stress/atom` command if you want per-atom pressure (stress). These per-atom values could be summed for a group of atoms via the `compute reduce`

So, we will take a look at what that basically means. So, the compute pressure basically calculates the entire pressure tensor. So, it has an ID, it has a group ID basically the total group of atoms over which you want to calculate that and say pressure and then you can say temp ID basically the temperature ID is the, the idea of the fix which basically calculates the temperature.

(Refer Slide Time: 10:38)

compute pressure command

Syntax

```
compute ID group-ID pressure temp-ID keyword ...
```

- ID, group-ID are documented in `compute` command
- pressure = style name of this compute command
- temp-ID = ID of compute that calculates temperature, can be NULL if not needed
- zero or more keywords may be appended
- keyword = ke or pair or bond or angle or dihedral or improper or kspace or fix or virial or pair/hybrid

Examples

```
compute 1 all pressure thermo_temp
compute 1 all pressure NULL pair bond
compute 1 all pressure NULL pair/hybrid 13/cut
```

Description

Define a computation that calculates the pressure of the entire system of atoms. The specified group must be "all". See the `compute stress/atom` command if you want per-atom pressure (stress). These per-atom values could be summed for a group of atoms via the `compute reduce` command.

The pressure is computed by the formula

$$P = \frac{Nk_B T}{V} + \frac{\sum_i^N r_i \cdot f_i}{dV}$$

where N is the number of atoms in the system (see discussion of DOF below), k_B is the Boltzmann constant, T is the temperature, d is the dimensionality of the system (2 or 3 for 2d/3d), and V is the system volume (or area in 2d). The second term is the virial, equal to $-dU/dV$, computed for all pairwise as well as 2-body, 3-body, 4-body, many-body, and long-range interactions, where r_j and f_j are the position and force vector of atom i, and the black dot indicates a dot product. When periodic boundary conditions are used, N' necessarily includes periodic image (ghost) atoms outside the central box, and the position and force vectors of ghost atoms are thus included in the summation. When periodic boundary conditions are not used, $N' = N$ = the number of atoms in the system. Fixes that impose constraints (e.g. the fix shake command) also contribute to the virial term.

A symmetric pressure tensor, stored as a 6-element vector, is also calculated by this compute. The 6 components of the vector are ordered xx, yy, zz, xy, xz, yz . The equation for the ij

Now, the expression for stress from statistical mechanics looks something like this, it has a temperature term. And this is the other term, which basically just depends on the potential energy of the system or the forces is the temperatures f naught there will only be due to the potential energy of the system.

$$P = \frac{Nk_B T}{V} + \frac{\sum_i^{N'} \tau_i \cdot f_i}{dV}$$

$$P_{IJ} = \frac{\sum_k^N m_k v_{kI} v_{kJ}}{V} + \frac{\sum_k^{N'} \tau_{kI} f_{kJ}}{V}$$

Student: Sir, the $\tau_i \cdot f_i$?

Professor: Yeah, $\tau_i \cdot f_i$.

(Refer Slide Time 11:01)

compute pressure command

compute cluster/atom command

compute chunk/spread/atom command

compute cluster/atom command

compute aggregate/atom command

compute cna/atom command

compute cnp/atom command

compute com command

compute com/chunk command

compute contact/atom command

compute coord/atom command

compute damage/atom command

compute dihedral command

compute dihedral/local command

compute dilatation/atom command

compute dipole/chunk command

compute dpd command

compute dpd/atom command

compute edpd/temp/atom command

compute entropy/atom command

compute erotate/asphere command

compute erotate/ligid command

compute erotate/sphere command

compute erotate/sphere/atom command

compute event/place command

compute fep command

computed for all pairwise as well as 2-body, 3-body, 4-body, many-body, and long-range interactions, where r_{ij} and f_{ij} are the position and force vector of atom i , and the black dot indicates a dot product. When periodic boundary conditions are used, N necessarily includes periodic image (ghost) atoms outside the central box, and the position and force vectors of ghost atoms are (thus) included in the summation. When periodic boundary conditions are not used, $N = N - 1$ - the number of atoms in the system. Fixes that impose constraints (e.g. the fix shake command) also contribute to the virial term.

A symmetric pressure tensor, stored as a 6-element vector, is also calculated by this compute. The 6 components of the vector are ordered xx, yy, zz, xy, xz, yz . The equation for the LJ components (where i and $j = xy, xz$) is similar to the above formula, except that the first term uses components of the kinetic energy tensor and the second term uses components of the virial tensor:

$$P_{IJ} = \frac{\sum_k^N m_k v_{kI} v_{kJ}}{V} + \frac{\sum_k^N r_{kI} f_{kJ}}{V}$$

If no extra keywords are listed, the entire equations above are calculated. This includes a kinetic energy (temperature) term and the virial as the sum of pair, bond, angle, dihedral, improper, kspace (long-range), and fix contributions to the force on each atom. If any extra keywords are listed, then only those components are summed to compute temperature or ke and/or the virial. The virial keyword means include all terms except the kinetic energy ke.

The pair/hybrid keyword means to only include contribution from a sub-style in a hybrid or hybrid/overlay pair style.

Details of how LAMMPS computes the virial efficiently for the entire system, including for many-body potentials and accounting for the effects of periodic boundary conditions are discussed in (Thompson).

NPTEL

LAMMPS

Search docs

1. Introduction

2. Install LAMMPS

3. Build LAMMPS

4. Run LAMMPS

5. Commands

6. Optional packages

7. Accelerate performance

8. Howto discussions

9. Example scripts

10. Auxiliary tools

11. Modify & extend LAMMPS

12. Use Python with LAMMPS

13. Errors

14. Building the LAMMPS manual

INDEX

Commands

Files

6 Computers

compute ackland/atom command

compute adf command

compute angle command

compute angle/local command

compute pressure command

Previous

Next

compute pressure command

Syntax

```
compute ID group-ID pressure temp-ID keyword ...
```

- ID, group-ID are documented in [compute command](#)
- pressure = style name of this compute command
- temp-ID = ID of compute that calculates temperature, can be NULL if not needed
- zero or more keywords may be appended
- keyword = ke or pair or bond or angle or dihedral or improper or kspace or fix or virial or pair/hybrid

Examples

```
compute 1 all pressure thermo temp
compute 1 all pressure NULL pair bond
compute 1 all pressure NULL pair/hybrid 12/cut
```

Description

Define a computation that calculates the pressure of the entire system of atoms. The specified group must be "all". See the [compute stress/atom](#) command if you want per-atom pressure (stress). These per-atom values could be summed for a group of atoms via the [compute reduce](#) command.

The pressure is computed by the formula

NPTEL

```

1 units          metal
2 atom_style     atomic
3 boundary       p p p
4
5 #FCC unit cell
6 variable a index 3.986
7 lattice fcc $(a)
8 region         myreg block    0 10 &
9                0 10 &
10               0 20
11 #Lattice units is used, which is the default
12 #In the argon example, we used box units.
13 create_box    1 myreg
14 create_atoms  1 region myreg
15 # Minimize using Embedded atom method potential for Al
16
17 pair_style     eam
18 pair_coeff     * * Al_inp.eam
19 minimize 1e-25 1e-25 100000 100000
20 #=====
21 #Total time is 10ps
22 #Total strain is 0.02
23 #Strain rate is 0.02 per ps, which is 0.002*1012 strains per second
24 #1010 strains per second!! Very high
25 #=====
26 timestep 0.001
27 reset_timestep 0
28 compute str all pressure NULL virial
29 variable i1 equal i2
30 variable ezz equal (i2-$i1)/$i1
31 fix defr all deform 1 z delta 0.0 100944 units box
32 dump dumpid all custom 100 deformto:virial.lammps:trj id x y z
33 thermo_style custom step time c_str[1] c_str[2] c_str[3] c_str[4] c_str[5] c_str[6] v_ezz lx ly
34 thermo 100
35 run 10000
36

```



So, this is the stress configuration. And this term as you can imagine, is associated with the velocities of the atom. And these velocities are 0 when you are actually calculating the elastic constants at 0 K. I do not want to calculate the temperature effects, just for the heck of it, just to see how good our calculations are without including those terms. And as you can imagine, metals and ceramics may not show a significant deviation from in their elastic constants unless in otherwise you are going to reach really high temperatures

So, we are interested only in this term for now. So, what I have to do is, it says here give the ID of the compute that calculates the temperature and can be null if not needed. So, I am going to say null, that is why I have given in that particular keyword, the word null. And then keyword, kinetic energy or pair or bond or angle or dihedral all these things is given. I am just going to give virial. So basically, the virial component of the stress alone is considered here. So basically, the second part is going to be considered here. See all the other terms actually do not make any sense for this particular case, you can if there are dihedral angles or if there are angle potentials associated with your interatomic potential, then you would be able to calculate stress components associated only with that potential for your system. For now, we just are going to give virial.

(Refer Slide Time: 12:36)

```
1 units metal
2 atom_style atomic
3 boundary p p p
4
5 #FCC unit cell
6 variable a index 3.986
7 lattice fcc $a)
8 region myreg block 0 10 &
9 0 10 &
10 0 20
11 #Lattice units is used, which is the default
12 ##in the argon example, we used box units.
13 create_box 1 myreg
14 create_atoms 1 region myreg
15 # Minimize using Embedded atom method potential for Al
16
17 pair_style eam
18 pair_coeff * * Al_jnp.eam
19 minimize 1e-25 1e-25 100000 100000
20 #=====
21 #Total time is 10ps
22 #Total strain is 0.02
23 #Strain rate is 0.02 per 10 ps, which is 0.02x10-11 strains per second
24 #2x10-9 strains per second!! Very high
25 #=====
26 timestep 0.001
27 reset_timestep 0
28 compute str all pressure NULL virial
29 variable il equal lz
30 variable ezz equal (lz-$il)/$il)
31 fix defz all deform 1 z delta 0.0 1.5944 units box
32 dump dumpid all custom 100 deformensile.lammpsstr id x y z
33 thermo_style custom step time c_str[1] c_str[2] c_str[3] c_str[4] c_str[5] c_str[6] v_ezz lx ly
34 thermo 100
35 run 10000
36
```



You could try what happens with other things, I have not tried it at this point. And then I am going to initialize the current length of the box `il`. And then calculate the strain to be `lz` minus `il` divided by `il`. So basically, the current length minus the initial length divided by the initial length would give me the strain. And then I set `fix`, this is the `fix` ID, the ID for the `fix`, this is the group of all the atoms over which I want to apply this `fix` and they set `deform`. And then I gave how frequently I want to deform it. So, every how many steps do I want to perform this deformation. So, I say `1` there. Can you all see this? Yes.

And then I say in the `z` direction and the `delta` by how much do I want to extend the `z` and the `z` direction. So, that is given by the keyword `delta`. And then the first and the second terms tell me how much in the `z` low, do I want to increase and how much in the `z` high, do I want to increase the length of this simulation box. So, that is `0.0` for `z` low and say `1.59444`, is `z` high. So, `1.5944` turns out to be about 2 percent strain. And then I set all these units distance units is in the box units, that means it is in angstroms.

And then I dumped all the coordinates as it is performing this deformation. And I also print out the output of the `compute` command using the `C` underscore `str1` it is a array, because it is an array, it is a pressure, it is a pressure is going to be an array with 6 components, because it calculates the pressure tensor. So, the `compute` will have in `str` 6 components, this is the `xx` component, this is the `yy`, `zz`, `xy`, `xz` and `yz` components. And then I am printing a variable, which is a strain for that as it is deforming, I want to print the variable.

And I also have just plotted the x and y just to show you that, that is actually not changing when I am actually performing this test. As I run this for about 10000 what, what this means is this deformation of 1.5944 will be achieved in about 10000 steps. So, if you look at this calculation carefully, this strain is about 0.02 and the strain rate is 0.02 per picosecond. So, 1234 so by not per picosecond by per 10s of picoseconds, in 10s of picoseconds because we have 10000 here. So, 10000 multiplied by 0.00 1234, so 10s of picosecond.

And this turns out to be 0.02 and 10 to the power 11 strains per second. Is that right? And that is a huge strain rate. So, I the reason why I made this simple calculation here is to show you that because of the timescales and the length scales that are actually involved in molecular dynamic simulations, whenever you perform such deformation experiments.

Student: Not the per second, it is still per picosecond.

Professor: Is it?

Student: Sir 0.02 per 10 picoseconds.

Professor: Yeah. So, which is which is 0.02 times, so 10 picoseconds is 10 into 10 to the power minus 12.

Student: Seconds.

Professor: Seconds, so, it goes the numerator and you get 100 to the power 11. So, I think this is correct. So, 2 into 10 to the power 9 strains per second, so huge strain rate. So, so, these calculations you can for, for the sake of calculating elastic constants, this might not be very important. But when you are talking about plasticity when there is deformation beyond the elastic regime, what will happen is you will, your, your stress strain plot is a very strong function of the rate at which you are actually pulling this thing, you must have studied these effects a lot in your material science.

So, whenever you are interpreting results in molecular dynamic simulations, especially those pertaining to these highs, plasticity and yielding and all that, you need to interpret your results very carefully. And if you are going to do realistic strain rates, your simulations will never complete within a reasonable time. So, you have to know how to interpret your results, at least qualitatively when you are performing these experiments. So, that is something that you need to keep in mind.

(Refer Slide Time: 17:45)

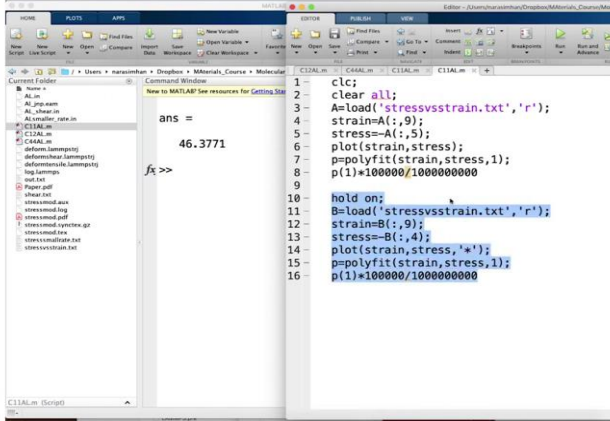
```
81    8000    39.86    8    -11868.829    -11868.829    -16181.987    8.7967687e-12    4.8584438e-12    7.2254688e-11    0
82    .0162    39.86    8.1    -12822.862    -12822.862    -16388.264    1.8238692e-11    4.922886e-11    4.7591691e-11    0.
83    8164    39.86    8.2    -12175.879    -12175.879    -16514.321    9.2789854e-12    5.9818282e-11    3.2327235e-11    0.
84    8164    39.86    8.3    -12327.88    -12327.88    -16728.159    1.857484e-12    3.9211882e-11    -3.3825846e-11    0.
85    8400    39.86    8.4    -12488.465    -12488.465    -16925.778    2.2838532e-12    -1.8681362e-11    5.8747083e-11    0
86    .0168    39.86    8.5    -12632.835    -12632.835    -17131.178    6.342551e-12    5.9112613e-11    9.3683779e-11    0.
87    8172    39.86    8.6    -12784.989    -12784.989    -17336.36    9.9751187e-12    6.3849547e-11    9.1976868e-11    0.
88    8700    39.86    8.7    -12936.929    -12936.929    -17541.326    -0.2413928e-12    -5.32626e-11    -7.4876272e-11    0.
89    8174    39.86    8.8    -13088.655    -13088.655    -17746.877    1.9685642e-12    3.846942e-11    6.8485843e-12    0.
90    8176    39.86    8.9    -13240.168    -13240.168    -17958.613    -8.1682458e-12    3.1799852e-11    -5.7189593e-11    0
91    8178    39.86    9    -13391.667    -13391.667    -18184.935    1.7245323e-11    -6.1161777e-11    -2.3288511e-11    0
92    8018    39.86    9.1    -13542.553    -13542.553    -18389.845    6.4886177e-12    1.8878797e-10    2.3988233e-11    0.
93    8182    39.86    9.2    -13693.427    -13693.427    -18562.943    1.3396863e-12    3.6614358e-11    -1.8597346e-11    0
94    .0184    39.86    9.3    -13844.889    -13844.889    -18766.63    1.8569587e-11    -3.93615e-11    1.9916424e-11    0.0
95    186    39.86    9.4    -13994.54    -13994.54    -18978.188    1.2183912e-11    -2.8113924e-11    -7.3126735e-11    0
96    8188    39.86    9.5    -14144.779    -14144.779    -19173.378    4.4184606e-12    -1.586853e-11    3.2377385e-11    0
97    .019    39.86    9.6    -14294.888    -14294.888    -19376.439    2.6487895e-12    7.4582384e-11    -9.9264452e-11    0
98    .0192    39.86    9.7    -14444.627    -14444.627    -19579.295    1.5676278e-11    9.8618842e-11    -3.1486331e-11    0
99    .0194    39.86    9.8    -14594.237    -14594.237    -19781.944    1.8188863e-12    4.869196e-11    1.9284135e-11    0
100    8196    39.86    9.9    -14743.637    -14743.637    -19984.389    1.7898864e-11    -1.8879811e-11    9.3136335e-11    0
101    .0198    39.86    10    -14893.829    -14893.829    -20186.629    -6.7722547e-13    3.5734271e-11    3.568898e-11    0
102    82    39.86    39.86
```



```
1 units          metal
2 atom_style     atomic
3 boundary       p p p
4
5 #FCC unit cell
6 variable a index 3.986
7 lattice fcc $a)
8 region         myreg block    0 10 &
9                0 10 &
10               0 20
11
12 #Lattice units is used, which is the default
13 #In the argon example, we used box units.
14 create_box     1 myreg
15 create_atoms   1 region myreg
16 # Minimize using Embedded atom method potential for Al
17
18 pair_style      eam
19 pair_coeff      * * Al_jmp.eam
20 minimize 1e-25 1e-25 100000 100000
21 #Total time is 180ps
22 #Total strain is 0.82
23 #Strain rate is 0.82 per 10 ps, which is 0.82x10-11 strains per second
24 #2x10-9 strains per second!! Very high
25 #=====
26 timestep 0.001
27 reset_timestep 0
28 compute str all pressure NULL virial
29 variable il equal 12
30 variable eez equal (l2-${il})/${il}
31 fix defz all deform 1 z delta 0.0 1.5944 units box
32 dump dumpid all custom 100 deformtensor.lmpstrj id x y z
33 thermo_style custom step time c_str[1] c_str[2] c_str[3] c_str[4] c_str[5] c_str[6] v_ezz lx ly
34 thermo 100
35 run 18000
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164

```

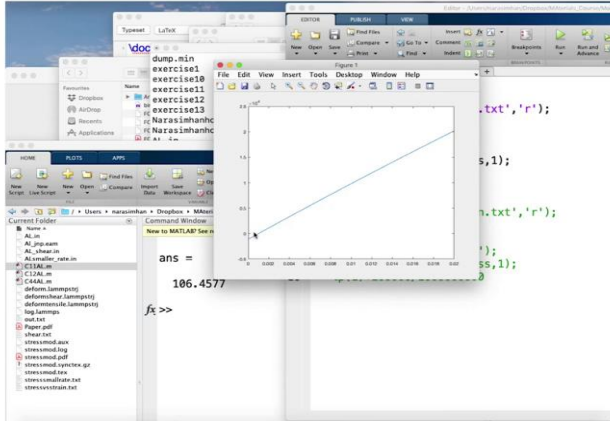

(Refer Slide Time: 18:28)



The screenshot shows the MATLAB environment with a script being executed. The Command Window displays the result: `ans = 46.3771`. The Editor window shows the following code:

```
1 clc;
2 clear all;
3 A=load('stressvsstrain.txt','r');
4 strain=A(:,9);
5 stress=A(:,5);
6 plot(strain, stress);
7 p=polyfit(strain, stress, 1);
8 p(1)*100000/1000000000
9
10 hold on;
11 B=load('stressvsstrain.txt','r');
12 strain=B(:,9);
13 stress=B(:,4);
14 plot(strain, stress, '+');
15 p=polyfit(strain, stress, 1);
16 p(1)*100000/1000000000
```


Below the MATLAB window, a man is seated at a desk with a laptop, looking at the screen.



The screenshot shows the MATLAB environment with a plot and a script execution. The Command Window displays the result: `ans = 106.4577`. The Editor window shows the following code:

```
1 clc;
2 clear all;
3 A=load('stressvsstrain.txt','r');
4 strain=A(:,9);
5 stress=A(:,5);
6 plot(strain, stress);
7 p=polyfit(strain, stress, 1);
8 p(1)*100000/1000000000
9
10 hold on;
11 B=load('stressvsstrain.txt','r');
12 strain=B(:,9);
13 stress=B(:,4);
14 plot(strain, stress, '+');
15 p=polyfit(strain, stress, 1);
16 p(1)*100000/1000000000
```

The plot shows a linear relationship between strain (x-axis) and stress (y-axis). Below the MATLAB window, a man is seated at a desk with a laptop, looking at the screen.



The MATLAB interface shows a plot of a straight line with a positive slope. The command window contains the following text:

```
ans =
    106.4577
fx >>
```

The NPTEL logo is visible in the bottom right corner of the composite image.

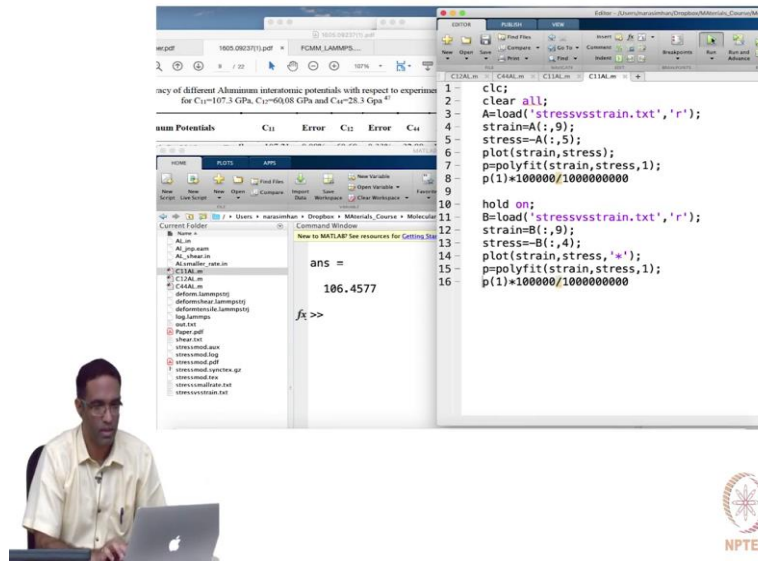
So, when I read this file and plot it, exercise 14, so I am going to first, somewhat I get a straight line however, you see that it is not really starting at 0, but there is there is a 0 stress state and a some sort of a positive strain value. So, these are some of the issues that you actually need to fix before you actually start calculating your stresses. But let us see what happens when you do this. When I did this, the slope of this line turned out to be 106.4577 gigapascals.

(Refer Slide Time: 19:26)

The PDF document displays the following table:

Aluminum Potentials	C ₁₁	Error	C ₁₂	Error	C ₄₄	Error
Mishin Ni-Al Co-2013.eam.alloy ⁴⁸	107.21	0.08%	60.60	0.33%	32.88	16.18%
AP9.eam.alloy ⁴⁴	107.03	0.25	61.06	0.42	31.05	9.72
Mishin NiAlD09 ⁴⁷	107.5	0.19	61.25	0.74	33.2	17.31
Mishin Al-Co-2013.eam.alloy ⁴¹	107.58	0.26	61.82	1.68	32.88	16.18
Mg-Al-se.eam.alloy ⁴⁶	107.08	0.21	58.35	4.03	31.6	11.66
Al Mg.eam.fs ⁴⁶	103.86	3.21	63.8	4.93	30.48	7.70
AlPb-seFe.eam.alloy ⁴⁹	105.63	1.56	58.2	4.28	31.85	12.54
ZnFe-Ti-Al-2003.eam.alloy ⁴⁸	98.27	8.42	56.23	7.52	26.02	8.06
AlB3.eam.alloy ⁴⁸	98.2	8.48	55.88	8.09	26.42	6.64
NiAl.eam.alloy ⁴⁹	100.52	6.32	54.65	10.12	26.14	7.63
AlFe-mn ²⁶	95.25	11.23	55.82	8.19	30.28	7.00
Al Fe.eam.fs ⁵⁰	94.15	12.26	55.18	9.24	30.28	7.00
Almn ²⁴	103.95	3.12	63.96	5.20	36.7	29.68
Al.eam.fs ⁵²	101.53	5.38	64.5	6.09	36.67	29.58
Al_jnp.eam ⁵⁰	96.58	9.99	74.61	22.71	40.43	42.86
Al-Li.eam.alloy ⁷¹	123.09	14.72	64.13	5.56	39.5	39.58
AlO.eam.alloy ⁶¹	82.92	22.72	69.3	13.98	30.37	7.31
Al_zhou.eam.alloy ⁴⁸	81.38	24.16	58.12	4.41	20.15	28.80
CoAl.eam.alloy ⁷²	83.16	22.50	55.1	9.37	21.41	24.35
AlCu.eam.alloy ⁷⁰	73.97	31.06	59.24	2.57	51.1	80.57
Farkas_Ni-Ti-Al_1996.eam.alloy ⁷³	171.64	59.96	116.1	93.24	91.83	224.49
Alset ⁴²	61.16	43.00	45.74	24.77	10.86	61.63
NiAlH_jca.eam.fs ⁷⁴	58.26	45.70	33.5	44.90	58.26	105.87
NiAlH_jca.eam.alloy ⁷⁴	58.98	45.03	32.9	45.89	58.98	108.41

The NPTEL logo is visible in the bottom right corner of the composite image.



So, let us check that paper and see what values they have. This is a very crude method, this actually takes a few seconds to run, it does not even take, it is not a long simulation. It is a very very quick simulation. So so, you do not have to expect the exact value turning out here, 96.58 is the value that these guys are getting. And the actual value is 107.3 gigapascal from experiment, so we are actually closer to experiments and what this and what this paper is telling us. Now what happens to C12, the C12 is also calculated in a very similar manner except that I need to, 79.971 is what I get and according to this paper, it is 74.61. So, in this we are quite far away from the experiments but closer to this value. At least it is in the same ballpark, at least it is in the same order of magnitude. From really, really simple calculations, something that runs very, very fast. So, that is C11 and C12.

(Refer Slide Time: 20:37)

Aluminum Potentials	shear.txt	stressvsstrain.txt
Mishin Ni Al Co 2013.eam.alloy ⁴¹	Narasimhanhomeair:exercise14 narasimhan\$ vim AL.in	Narasimhanhomeair:exercise14 narasimhan\$ vim stressvsstrain.txt
AP9.eam.alloy ⁴⁴	Narasimhanhomeair:exercise14 narasimhan\$ vim AL.in	Narasimhanhomeair:exercise14 narasimhan\$ vim log_lammps
Mishin NiAl2009 ⁴²	Narasimhanhomeair:exercise14 narasimhan\$ ls	Narasimhanhomeair:exercise14 narasimhan\$ vim stressvsstrain
Mishin Al Co 2013.eam.alloy ⁴¹	AL.in	Paper.pdf
Mg Al set.eam.alloy ⁴⁵	AL_shear.in	stressmod.log
AI Mg.eam.f ⁴⁶	Al_smaller_rate.in	deformshear.lampstrj
AI Pb set.f.eam.alloy ⁴⁷	Al_jnp.eam	deformtensile.lampstrj
Zope Ti Al 2003.eam.alloy ⁴⁸	C12AL.m	stressmod.synctex.gz
AB13.eam.alloy ⁴⁹	C12AL.m	log_lammps
NIAl.eam.alloy ⁴⁹	C44AL.m	out.txt
AI Fe.eam.f ⁷⁹	Narasimhanhomeair:exercise14 narasimhan\$ cd AL	shear.txt
Alum ⁵⁰	AL.in	AL_shear.in
AI Fe.eam.f ⁷⁹	Narasimhanhomeair:exercise14 narasimhan\$ cd AL.in	Al_smaller_rate.in
AI.eam.f ⁷³	-bash: cd: AL.in: Not a directory	Narasimhanhomeair:exercise14 narasimhan\$ cd AL.in
AI.jnp.eam ⁵⁹	Narasimhanhomeair:exercise14 narasimhan\$ vim AL_d	-bash: cd: AL_shear.in: Not a directory
AI LE.eam.alloy ⁷¹	123.09 1472 6418 556 39.5 39.58	Narasimhanhomeair:exercise14 narasimhan\$ vim AL_d
AI O.eam.alloy ⁷¹	82.92 22.72 69.3 13.98 30.37 7.31	
AI O.ho.eam.alloy ⁴⁸	81.38 24.16 58.12 4.41 20.15 28.80	
CoAl.eam.alloy ⁷²	83.16 22.50 55.1 9.37 21.41 24.35	
AI Cu.eam.alloy ³⁵	73.97 31.06 59.24 2.57 51.1 80.57	
Faruk Nb Ti Al 1996.eam.alloy ⁷³	171.64 39.90 116.1 93.24 91.83 224.49	
AI set ⁶⁵	61.16 43.00 45.74 24.77 10.86 61.63	
NIAlH.je.eam.f ⁷⁴	58.26 45.70 33.5 44.90 58.26 105.87	
NIAlH.je.eam.alloy ⁷⁴	58.98 45.03 32.9 45.89 58.98 108.41	

```
4
5 #FCC unit cell
6 variable a equal 3.986
7 #for shear deformation, we need a triclinic box
8 #lattice custom $(a) a1 1.0 0.0 0.0 a2 0.0 1.0 0.0 a3 0.0 0.0 1.0 basis 0.0 0.0 0.0 basis 0.5 0.5 0.0 basis 0.5
9 0.0 0.5 basis 0.0 0.5 0.5
10 region myrec prism 0 10 &
11 0 10 &
12 0 0 0
13 #Lattice units is used, which is the default
14 #in the argon example, we used box units.
15 create_box 1 myrec
16 create_atoms 1 region myrec basis 1 1 basis 2 1 basis 3 1 basis 4 1
17 # Minimize using Embedded atom method potential for Al
18
19 pair_style eam
20 pair_coeff ** Al_jnp.eam
21 minimize 1e-25 1e-25 100000 100000
22 #Total time is 10ps
23 #Total strain is 0.02
24 #Strain rate is 0.02 per ps, which is 0.002x10^12 strains per second
25 #2x10^9 strains per second!! Very high
26 #
27 #
28 timestep 0.001
29 reset_timestep 0
30 compute str all pressure NULL virial
31 variable ily equal ly
32 variable ilx equal lx
33 variable ilz equal lz
34 variable exy equal xy/${ily}
35 variable exz equal xz/${ilz}
36 variable eyz equal yz/${ily}
37 variable exx equal (lx-${ilx})/${ilx}
38 variable eyy equal (ly-${ily})/${ily}
39 variable ezz equal (lz-${ilz})/${ilz}
40 fix defxy all deform 1 xz delta 2.0898 units box
41 dump dumpid all custom 200 deformahear.lampstrj id x y z
42 thermo_style custom c_str[1] v_exe c_str[2] v_eyy c_str[3] v_ezz c_str[4] v_ezy c_str[5] v_ezx c_str[6] v_eyz
43 thermo 100
44 run 10000
45
```

The next thing is applying shear. So, that is that in the, AL underscore. So that is the calculation that, that's needs to be done for a shear case, for a shear deformation, the same fix deform command is going to be used here. So, I am going to be talking about certain important lines and then going back here and, and tell you why I actually did it, did it this way. If you notice here, here, the Lattice structure is not defined just as Lattice FCC dollar a. I am defining it, like what I did for other crystal structures.

And that is because my, my simulation box needs to be a triclinic box if I want to be able to tilt it in LAMMPS. So, if the fixed deform command has to have a xy, so previously, we had an z here, telling that I want the z part, the z direction to be deforming for the simulation box. Now I am saying the x,y needs to be deforming. So, if that has to happen, then what I am

supposed to give you. So, imagine, imagine this to be the x axis and this to be the y axis, what I am supposed to be giving is the distance by which y needs to be moved towards x, so basically, this distance, that is what I am supposed to be giving in order to deform the simulation box and share it. So, xy LAMMPS means distance y needs to be moved in the direction of x, that is what it means. xz means distance z needs to be moved in the direction of x.

(Refer Slide Time: 22:40)

The screenshot shows the LAMMPS website with the following content:

- Commands:**
 - fix adapt command
 - fix adapt/foam command
 - fix addforce command
 - fix addtorque command
 - fix append/atoms command
 - fix atc command
 - fix atom/fix command
 - fix ave/all command
 - fix ave/chunk command
 - fix ave/correlate command
 - fix ave/correlate/foam command
 - fix ave/histo command
 - fix ave/histo/weight command
 - fix ave/time command
 - fix aveforce command
 - fix balance command
- fix command description:**
 - ID, group-ID are documented in fix command
 - deform = style name of this fix command
 - N = perform box deformation every this many timesteps
 - one or more parameter/arg pairs may be appended
 - parameter = x or y or z or xy or xz or yz
 - #, P, arg = style values
 - angle = final or delta or scale or vel or erate or trate or volume or wiggle or var
 - final value = la bl
 - bl = box boundaries at end of run (distance units)
 - delta value = dln dhl
 - dln dhl = change in box boundaries at end of run (distance units)
 - scale value = factor
 - factor = multiplicative factor for change in box length at end of run
 - vel value = V
 - V = change box length at this velocity (distance/time units), effectively an engineering shear strain rate
 - erate value = E
 - E = engineering strain rate (1/time units)
 - trate value = R
 - R = true strain rate (1/time units)
 - volume value = none = adjust this die to preserve volume of system
 - wiggle value = A Tp
 - A = amplitude of oscillation (distance units)
 - Tp = period of oscillation (time units)
 - variable value = v_name1 v_name2
 - v_name1 = variable with name1 for box length change as function of time
 - v_name2 = variable with name2 for change rate as function of time
 - xy, xz, yz arg = style value
 - angle = final or delta or vel or erate or trate or wiggle
 - final value = tilt
 - tilt = tilt factor at end of run (distance units)
 - delta value = dtilt
 - dtilt = change in tilt factor at end of run (distance units)
 - vel value = V
 - V = change tilt factor at this velocity (distance/time units), effectively an engineering shear strain rate
 - erate value = E
 - E = engineering shear strain rate (1/time units)
 - trate value = R
 - R = true shear strain rate (1/time units)

The screenshot shows a terminal window with the following content:

```

Icon?
Important_Argon_Paper_2017.pdf
SI.sw
dump.eqm1
dump.eqm2
dump.eqm3
dump.min
exercise1
exercise10
exercise11
exercise12
exercise13
exercise7
exercise8
exercise8.1
exercise9
exerciserdf
exerciserdfgas
in.threobody
input1_argon.in
log.lammps
log.n_ar_1000.txt
logfile.txt
output.out
Narasimhanhomeair:Argon narasimhan$ cd exercise14/
Narasimhanhomeair:exercise14 narasimhan$ ls
AL.in Paper.pdf stressmod.aux
AL_shear.in deform.lammpstrj stressmod.log
AL_smaller_rate.in deformshear.lammpstrj stressmod.pdf
AL_jnp_eam deformtensile.lammpstrj stressmod.synctex.gz
C12AL.m log.lammps stressmod.tex
C12AL.m out.txt stresssmallrate.txt
C64AL.m shear.txt stressvsstrain.txt
Narasimhanhomeair:exercise14 narasimhan$ vim AL.in
Narasimhanhomeair:exercise14 narasimhan$ vim AL.in
Narasimhanhomeair:exercise14 narasimhan$ vim log.lammps
Narasimhanhomeair:exercise14 narasimhan$ vim stress
stressmod.aux stressmod.pdf stressmod.tex stressvsstrain.txt
stressmod.log stressmod.synctex.gz stresssmallrate.txt
Narasimhanhomeair:exercise14 narasimhan$ vim stressvsstrain.txt
Narasimhanhomeair:exercise14 narasimhan$ ls
AL.in Paper.pdf stressmod.aux
AL_shear.in deform.lammpstrj stressmod.log
AL_smaller_rate.in deformshear.lammpstrj stressmod.pdf
AL_jnp_eam deformtensile.lammpstrj stressmod.synctex.gz
C12AL.m log.lammps stressmod.tex
C12AL.m out.txt stresssmallrate.txt
C64AL.m shear.txt stressvsstrain.txt
Narasimhanhomeair:exercise14 narasimhan$ cd AL
AL.in AL_shear.in AL_smaller_rate.in
Narasimhanhomeair:exercise14 narasimhan$ cd AL.in
-bash: cd: AL.in: Not a directory
Narasimhanhomeair:exercise14 narasimhan$ cd AL_shear.in
-bash: cd: AL_shear.in: Not a directory
Narasimhanhomeair:exercise14 narasimhan$ vim AL_shear.in
Narasimhanhomeair:exercise14 narasimhan$ cd AL_shear.in
  
```



So, you will find these things in the manual. And it can be pretty confusing the first time you look at it. So, the tilt, the extent of tilt that needs to be given to these axis, is what is being given here. So, let us take a look at the fix deform command. So, the fix deform command takes an ID, a group ID, all here deform is the keyword of the style, sorry, the name of this command, n is every step and then parameter can either be x, y,z, xy, xz or yz. So, what that means is, xy means like I said, you need to give the value of the tilt, the distance, this this second thing, axis needs to be moved towards the first variable, that is important to remember.

So, how do you figure that out? So that also needs to be in angstroms. So, what you can do is, so you take your MATLAB, and then you say that say you want a 3 degree tilt. So $\tan d 3$ degrees multiplied by this y will actually give you this distance. So, this y turns out to be

39.86 in our case, which is 2.0890. So, so, that is the reason why I have here 2.0890 units is box again and this and, what is that?

Student: Low and high, I mean 0.

Professor: No, there is low and high there is just the total amount of tilt that you are to give for the xy, only for x y and z do you need to give a how much you want to pull, push down in the x, how much you want to pull up in the xy, how much you want to increase that. But that also you see, I gave this to be 0 and just increase this. So, it is the same thing. They are just giving you an option to do that if in case that is required. So...

Student: Sir, 0 minus this thing, will it give an error?

Professor: Yes, it gave an error. Yes, that is right. You cannot put 0 here, because delta, for example, if you see x here, Delta is d low and d high, whereas xy is just the tilt, it is just a tilt.

(Refer Slide Time: 24:59)

```
4
5 #FCC unit cell
6 variable a equal 3.986
7 #For shear deformation, we need a triclinic box
8 lattice custom $(a) at 1.0 0.0 0.0 a2 0.0 1.0 0.0 a3 0.0 0.0 1.0 basis 0.0 0.0 0.0 basis 0.5 0.5 0.0 basis 0.5
9 0.0 0.5 basis 0.0 0.5 0.5
10 region myreg prism 0 10 &
11 0 10 &
12 0 0 0
13 #Lattice units is used, which is the default
14 #In the argon example, we used box units.
15 create_box 1 myreg
16 create_atoms 1 region myreg basis 1 1 basis 2 1 basis 3 1 basis 4 1
17 # Minimize using Embedded atom method potential for Al
18
19 pair_style eam
20 pair_coeff * = Al_fcc.eam
21 minimize 1e-25 1e-25 100000 100000
22 #=====
23 #Total time is 10ps
24 #Total strain is 0.02
25 #Strain rate is 0.02 per ps, which is 0.002x10(12) strains per second
26 #2x10(9) strains per second!! Very high
27 #=====
28 timestep 0.001
29 reset_timestep 0
30 compute str all pressure NULL virial
31 variable lly equal ly
32 variable llx equal lx
33 variable llz equal lz
34 variable exy equal xy/$lly)
35 variable exz equal xz/$llz)
36 variable eyz equal yz/$llz)
37 variable exx equal (lx-$llx)/$lly)
38 variable eyy equal (ly-$lly)/$lly)
39 variable ezz equal (lz-$llz)/$llz)
40 fix defxy all deform 1 xy delta 2.0890 units box
41 thermo_style custom 200 deformshear.lammpstrj id x y z
42 thermo_style custom c_str[1] v_exx c_str[2] v_eyy c_str[3] v_ezz E_str[4] v_exy c_str[5] v_ezz c_str[6] v_eyz
43 thermo 100
44 run 10000
45
```



```

Narasimhanhomeair:exercisel4 narasimhan$ ls
Al.in          C464L.m          out.txt          stressmod.tex
Al_shear.in   Paper.pdf        shear.txt        stresssmallrate.txt
Alsmaller_rate.in  deform.lampstrj stressmod_aux    stressvsstrain.txt
Al_jnp.eam    deformshear.lampstrj stressmod.log
C11AL.m       deformtensile.lampstrj stressmod.pdf
C11AL.m       log.lammps      stressmod_synctex.gz
Narasimhanhomeair:exercisel4 narasimhan$ vim log.lammps
Narasimhanhomeair:exercisel4 narasimhan$ ls
Al.in          C464L.m          out.txt          stressmod.tex
Al_shear.in   Paper.pdf        shear.txt        stresssmallrate.txt
Alsmaller_rate.in  deform.lampstrj stressmod_aux    stressvsstrain.txt
Al_jnp.eam    deformshear.lampstrj stressmod.log
C11AL.m       deformtensile.lampstrj stressmod.pdf
C11AL.m       log.lammps      stressmod_synctex.gz
Narasimhanhomeair:exercisel4 narasimhan$ cd ../exercisel5/
Narasimhanhomeair:exercisel5 narasimhan$ ls
Al.in          C11AL.m          deformtensile.lampstrj strain.txt
Al_jnp.eam     DELTz.txt       log.lammps
Narasimhanhomeair:exercisel5 narasimhan$ vim log.lammps
Narasimhanhomeair:Argon narasimhan$ ls
Ar_1000_ar_0.txt      dump.min          exercise7
Argon.Example_Inputs exercise1         exercise8
Argon.Example_Inputs.zip exercise10        exercise8.1
Denspak3.1           exercise11        exercise9
Exercise_list.doc    exercise12        exerciserdf
Exercise_list.odt    exercise13        exerciserdfgas
FCC_Argon.in         exercise14        in.threebody
Icon?                exercise15        input1_argon.in
Important_Argon_Paper_2017.pdf exercise2         log.lammps
Si.m                 exercise3         log-ar_1000.txt
dump.eqm1           exercise4         logfile.txt
dump.eqm2           exercise6         output.out
dump.eqm3           exercise6
Narasimhanhomeair:Argon narasimhan$ cd exercisel4/
Narasimhanhomeair:exercisel4 narasimhan$ ls
Al.in          C464L.m          out.txt          stressmod.tex
Al_shear.in   Paper.pdf        shear.txt        stresssmallrate.txt
Alsmaller_rate.in  deform.lampstrj stressmod_aux    stressvsstrain.txt
Al_jnp.eam    deformshear.lampstrj stressmod.log
C11AL.m       deformtensile.lampstrj stressmod.pdf
C11AL.m       log.lammps      stressmod_synctex.gz
Narasimhanhomeair:exercisel4 narasimhan$ vim s

```



```

Alsmaller_rate.in  deform.lampstrj  stressmod_aux  stressvsstrain.txt
Al_jnp.eam         deformshear.lampstrj stressmod.log
C11AL.m            deformtensile.lampstrj stressmod.pdf
C11AL.m            log.lammps      stressmod_synctex.gz
Narasimhanhomeair:exercisel4 narasimhan$ vim shear.txt

```

1	183.9019	0	1183.9019	0	1183.9019	0	-6.2877096e-11	0	4.756
2	9212e-11	0	4.5697901e-12	0	0	0	0	0	0
3	1184.0913	0	1184.3842	0	0	0	0	0	0
4	57e-11	0	2.8880232e-11	0	0	0	0	0	0
5	1184.667	0	1185.5275	0	0	0	0	0	0
6	486e-12	0	2.1724409e-11	0	0	0	0	0	0
7	1185.6376	0	1187.5802	0	0	0	0	0	0
8	064e-11	0	2.6144635e-11	0	0	0	0	0	0
9	1187.0437	0	1118.5803	0	0	0	0	0	0
10	62e-11	0	1.7932986e-11	0	0	0	0	0	0
11	1188.8857	0	1114.296	0	0	0	0	0	0
12	166e-11	0	1.843846e-11	0	0	0	0	0	0
13	1111.1634	0	1118.9592	0	0	0	0	0	0
14	74e-11	0	2.2290015e-11	0	0	0	0	0	0
15	1113.8772	0	1124.4928	0	0	0	0	0	0
16	8e-11	0	2.6286425e-11	0	0	0	0	0	0
17	1117.027	0	1138.8968	0	0	0	0	0	0
18	537e-12	0	1.9294865e-11	0	0	0	0	0	0
19	1120.6111	0	1138.1697	0	0	0	0	0	0
20	38e-11	0	2.4699898e-11	0	0	0	0	0	0
21	1124.6288	0	1146.3107	0	0	0	0	0	0
22	239e-11	0	1.8574427e-11	0	0	0	0	0	0
23	1129.0815	0	1155.3214	0	0	0	0	0	0
24	11e-12	0	-1.3418668e-12	0	0	0	0	0	0
25	1133.9698	0	1165.2825	0	0	0	0	0	0
26	51e-11	0	2.1278973e-11	0	0	0	0	0	0
27	1139.2939	0	1176.9543	0	0	0	0	0	0
28	375e-13	0	2.9326214e-11	0	0	0	0	0	0
29	1145.0539	0	1187.5771	0	0	0	0	0	0
30	684e-11	0	3.0987222e-12	0	0	0	0	0	0
31	1151.2498	0	1280.8712	0	0	0	0	0	0
32	778e-12	0	-8.1101838e-14	0	0	0	0	0	0
33	1157.8814	0	1213.4366	0	0	0	0	0	0
34	93e-11	0	1.6644608e-11	0	0	0	0	0	0
35	1164.4547	0	1227.6735	0	0	0	0	0	0
36	912e-13	0	-3.958191e-12	0	0	0	0	0	0
37	1172.4514	0	1242.7819	0	0	0	0	0	0
38	496e-11	0	1.1322659e-11	0	0	0	0	0	0



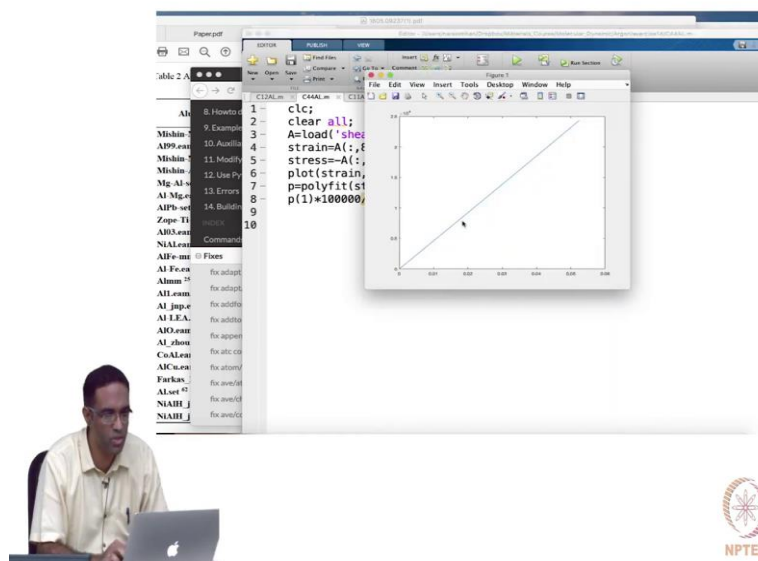
39	2455.7413	0	3857.6157	0	0	0	827.63164	0	-19411.531	0.041926744	-1.1808
40	433e-12	0	1.8016194e-12	0	0	0	828.70253	0	-19656.207	0.042450828	-1.8114
41	2489.6489	0	3927.114	0	0	0	813.68852	0	-19900.944	0.042974912	-1.35258
42	499e-12	0	7.1173807e-12	0	0	0	806.58939	0	-20145.751	0.043498991	-1.58747
43	2523.9258	0	3997.4389	0	0	0	799.40519	0	-20390.597	0.044023081	-1.38286
44	2558.5731	0	4068.592	0	0	0	792.13598	0	-20635.48	0.044547165	-4.7116
45	6e-11	0	4.3408528e-12	0	0	0	784.7818	0	-20880.405	0.045071249	-2.55312
46	2593.6283	0	4148.6886	0	0	0	777.3427	0	-21125.38	0.045595334	-2.98334
47	92e-12	0	4.8712011e-12	0	0	0	769.81766	0	-21370.409	0.046119418	-1.24213
48	2629.1802	0	4213.4976	0	0	0	762.20739	0	-21615.499	0.046643502	-8.35945
49	386e-13	0	1.488994e-11	0	0	0	754.5125	0	-21860.655	0.047167587	-8.83167
50	2644.9888	0	4287.2599	0	0	0	746.73889	0	-22185.883	0.047691671	-1.40897
51	8e-12	0	1.0124564e-11	0	0	0	738.86919	0	-22351.188	0.048215755	-1.0286
52	2781.2941	0	4361.8966	0	0	0	730.92887	0	-22596.577	0.048739839	-1.6992
53	05e-12	0	1.2929775e-11	0	0	0	722.88817	0	-22842.055	0.049263924	-1.32708
54	2738.0163	0	4437.4078	0	0	0	714.77138	0	-23087.628	0.049788008	-5.3623
55	96e-11	0	5.1808141e-12	0	0	0	706.57973	0	-23333.3	0.050312092	-1.3628
56	2775.1553	0	4513.7983	0	0	0	698.28611	0	-23579.877	0.050836177	-1.19378
57	91	2812.7111	0	4591.0686	0	0	689.91729	0	-23824.965	0.051360261	-2.59922
58	42e-13	0	1.25757e-11	0	0	0	681.4646	0	-24070.969	0.051884345	-1.5636
59	2860.6836	0	4609.2045	0	0	0	672.92813	0	-24317.095	0.05240843	-1.28373
60	77e-11	0	2.1278349e-11	0	0	0	0	0	0	0	0
61	2889.0725	0	4748.2278	0	0	0	0	0	0	0	0
62	592e-11	0	3.4617083e-11	0	0	0	0	0	0	0	0
63	94	2927.8776	0	4828.1314	0	0	0	0	0	0	0
64	766e-13	0	5.0881748e-12	0	0	0	0	0	0	0	0
65	2967.0992	0	4908.0167	0	0	0	0	0	0	0	0
66	59e-11	0	7.9455225e-12	0	0	0	0	0	0	0	0
67	3806.7375	0	4990.5856	0	0	0	0	0	0	0	0
68	763e-12	0	3.2386141e-11	0	0	0	0	0	0	0	0
69	3046.7925	0	5073.1399	0	0	0	0	0	0	0	0
70	545e-11	0	4.615783e-12	0	0	0	0	0	0	0	0
71	3887.2634	0	5156.5805	0	0	0	0	0	0	0	0
72	67e-12	0	3.549286e-11	0	0	0	0	0	0	0	0
73	3128.149	0	5246.9882	0	0	0	0	0	0	0	0
74	61e-11	0	2.0884993e-11	0	0	0	0	0	0	0	0
75	3169.4503	0	5326.1244	0	0	0	0	0	0	0	0
76	85e-11	0	2.9448569e-11	0	0	0	0	0	0	0	0
77	211.1672	0	5412.23	0	0	0	0	0	0	0	0
78	03e-11	0	1.0939933e-11	0	0	0	0	0	0	0	0



Again, this also is done over 10000 time steps. So, it will do it in about 10 Pico seconds. And I am printing out everything here. So, I am printing out the stress, the epsilon xx stress, epsilon yy, epsilon, sigma zz and the strain in the zz direction and all the correspondence and this is what we are expecting to have non 0 values. We are expecting the xy shear stress and the corresponding shear strain to have non 0 values which are the 1 2 3 4 5 6 7th and the 8th column.

So, if you take a look at the, so 1 2 3 4 5 6 7th and 8th column, so initially the strains are small. Oh no, this is not right. . No no, it is exercise 14 only. So, I just printed out all those values into another file and deleted all the unnecessary lines. So, if you look at the this is the 1st stress 1st strain, 2nd stress 2nd strain, 3rd stress 3rd strain, 4th stress and the corresponding value of strain initially is 0. But in the next line, if you see this becomes there is a non 0 value for these strains. And finally, you reach the require, required strain, you reach the required strain.

(Refer Slide Time: 27:15)



The image is a composite showing a man in a light-colored shirt sitting at a laptop. Overlaid on the scene are two windows from a MATLAB environment. The top window is a code editor displaying the following MATLAB script:

```
1 = clear all;
2 = A=load('shear');
3 = strain=A(:,1:4);
4 = stress=A(:,5:8);
5 = plot(strain, stress);
6 = p=polyfit(stress, strain);
7 = p(1)*1000000;
```

The bottom window is a plot titled 'Figure 1' showing a linear relationship between strain and stress. The x-axis is labeled 'strain' and ranges from 0 to 0.08. The y-axis is labeled 'stress' and ranges from 0 to 14. A blue line starts at the origin (0,0) and extends to approximately (0.08, 14). The plot includes a legend and a title bar.

ans =
46.3771

fx >> |



for C11=107.3 GPa, C12=60.08 GPa and C13=28.3 GPa

Aluminum Potentials	C11	Error	C12	Error	C13	Error
Mishin-Ni-Al-Co-2013.eam.alloy ⁴¹	107.21	0.08%	60.60	0.33%	32.88	16.18%
APP.eam.alloy ⁴⁴	107.03	0.25	61.06	6.43	31.05	9.72
Mishin-NiAl2007 ⁴²	107.5	0.19	61.25	0.74	33.2	17.31
Mishin-Al-Co-2013.eam.alloy ⁴¹	107.58	0.26	61.82	1.68	32.88	16.18
Mg-Al-vc.eam.alloy ⁴⁵	107.08	0.21	58.35	4.03	31.6	13.66
Al-Mg.eam.fs ⁴⁶	103.86	3.21	63.8	4.93	30.48	7.70
AlPb-vc.eam.alloy ⁴⁷	105.63	1.56	58.2	4.28	31.85	12.54
Zope-Ti-Al-2003.eam.alloy ⁴⁸	98.27	8.42	56.23	7.52	26.02	8.06
AlO3.eam.alloy ⁴⁹	98.2	8.48	55.88	8.09	26.42	6.64
NiAl.eam.alloy ⁴⁹	100.52	6.32	54.65	10.12	26.14	7.63
AlFe.eam.fs ⁵⁰	95.25	11.23	55.82	8.19	20.28	7.00
Al-Fe.eam.fs ⁵⁰	94.15	12.26	55.18	9.24	30.28	7.00
AlMn ⁵¹	103.95	3.12	63.96	5.20	26.7	29.68
AlH.eam.fs ⁵²	101.53	5.38	64.5	6.09	36.67	29.58
Al-jup.eam ⁵³	96.58	9.99	74.61	22.71	30.88	42.86
Al-LEA.eam.alloy ⁵⁴	123.09	14.72	64.18	5.56	39.5	39.58
AlO.eam.alloy ⁵⁴	82.92	22.72	69.3	13.98	30.37	7.31
Al-zhou.eam.alloy ⁴⁶	81.38	24.16	58.12	4.41	20.15	28.80
CuAl.eam.alloy ⁵⁵	83.16	22.50	55.1	9.37	21.41	24.55
AlCu.eam.alloy ⁵⁴	73.07	31.06	49.24	2.47	51.1	80.57
Farkas-Nb-Ti-Al-1996.eam.alloy ⁵⁷	171.64	59.96	116.1	93.24	91.83	224.49
Alset ⁴²	61.16	43.00	45.74	24.77	10.86	61.63
NIAlH-jca.eam.fs ⁵⁸	58.26	45.70	33.5	44.90	58.26	105.87
NIAlH-jca.eam.alloy ⁵⁸	58.98	45.03	32.9	45.89	58.98	108.41



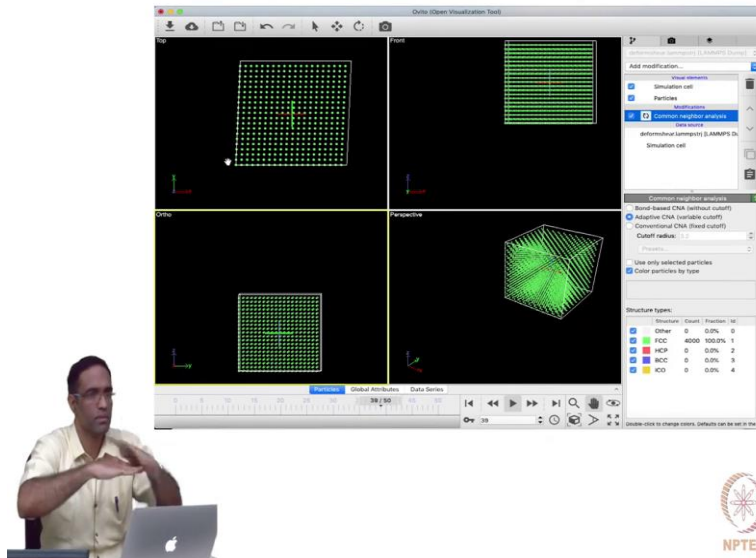
Simulation cell
Particles
Common neighbor analysis
Informational LAMMPS (LAMMPS) Simulation cell

Common neighbor analysis
Bond-based CNA (without cutoff)
Adaptive CNA (variable cutoff)
Conventional CNA (fixed cutoff)
Cutoff radius: ...

Use only selected particles
Color particles by type

Structure Types
Structure Count Fraction M
Other 0 0.0% 0
FCC 8000 100.0% 1
HCP 0 0.0% 2
BCC 0 0.0% 3
ICD 0 0.0% 4





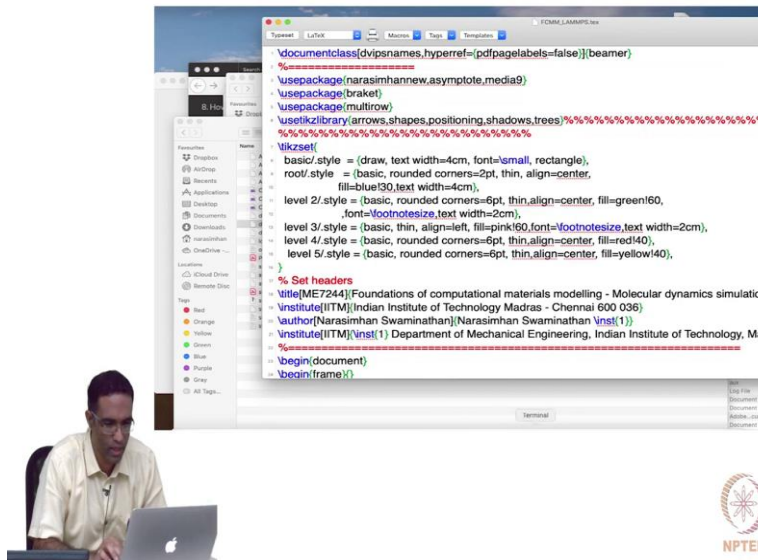
Again, if you plot this you get a straight line. So that is C11, C12 and this is just C44. And then if you look at the corresponding value of p_1 , it turns out to be 46.3771. Let us see how we are doing here, 40.43, the actual value is about 28.3. So, it is not it is not great, but it is okay as far as the potential is concerned. So, this potential is supposed to not do anything better than that. So so, you need to keep in mind what experiments are saying about your values and what this potential is capable of predicting. Now, this is not a very amazing thing, because most likely these embedded atom potentials are going to be fitted for actually reproducing some of these elastic constants.

So, but in case you have an interatomic potential for which you do not know the elastic constants to get the order of magnitude, you can do some simple tests like this to actually figure it out. And you have to make sure and you are you are making sure every time that the, you are looking only in the linear portion of the curve because when you are pulling, if you look at the extent of pulling that is going on here. I will show this to you later, you can actually hardly see it being moved, it is actually being pulled off. So, it is being pulled very little, so you cannot see it much. There is some little bit of moment if you observe very carefully.

Can you see the box move? Moved a little bit. No no, wait, once it loads completely it will appear to be a little bit fast. So it is more. Oh, I should switch this off, one minute. Now I think you should be able to see it move a little bit. And the shear thing also moves very little, but that is a little bit more can be seen a little bit more clearly. You can see the box being sheared in the xy, the y is moved, like what, what expected the (extend) extend, xy specifies

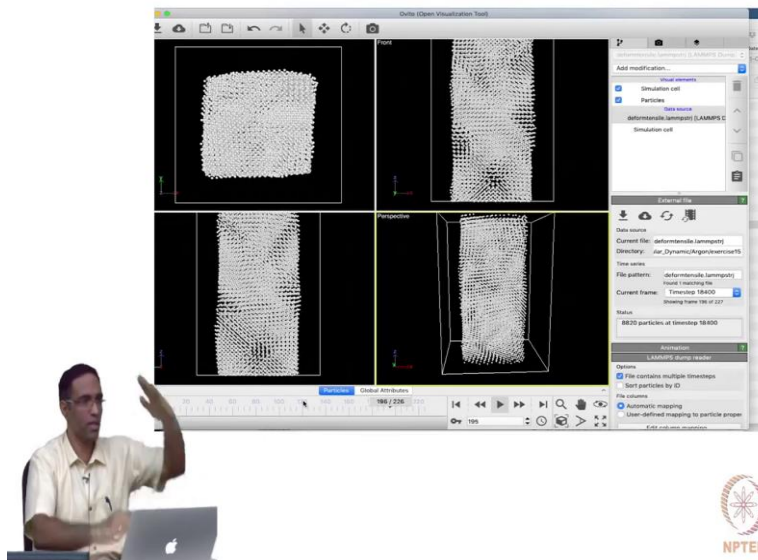
how much y is moved towards x . So, you see that, it is that axis which is moving towards y by this, by this amount. This information as to what xy , yz and xz means is also useful for constructing triclinic simulation boxes. So, we have not done that, but with that information, you should be able to do it pretty easily.

(Refer Slide Time: 30:22)



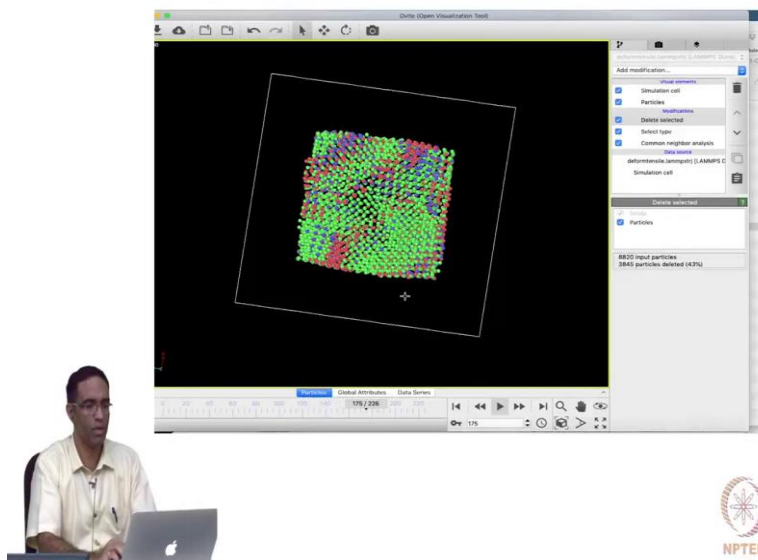
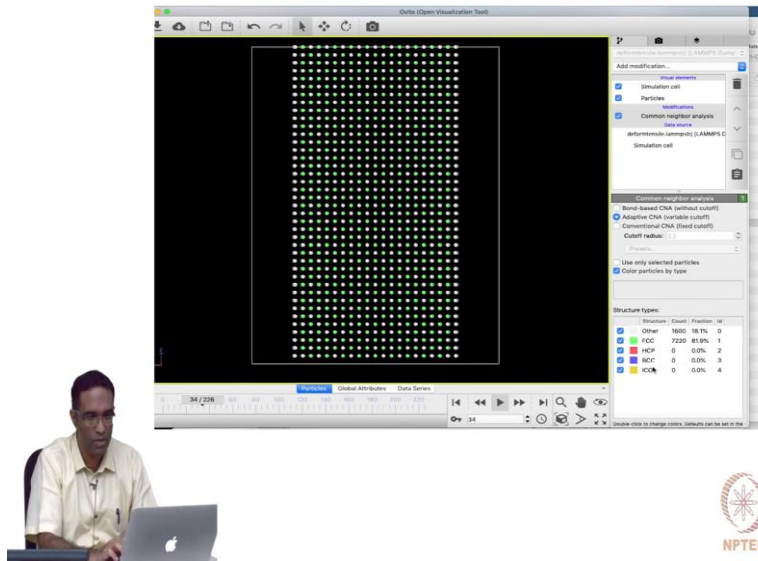
The image shows a man in a light-colored shirt sitting at a desk with a laptop. He is looking at a computer screen that displays LaTeX code for a Beamer presentation. The code includes package loading, style definitions for different levels, and header information. The code is as follows:

```
\documentclass[dvipsnames,hyperref={pdfpagelabels=false}]beamer
%=====
\usepackage[narasimhannew,asymptote,media9]
\usepackage[braket]
\usepackage[multitrow]
\usetikzlibrary[arrows,shapes,positioning,shadows,trees]%%%%%%%%%%
%
\tikzset{
  basic/.style = (draw, text width=4cm, font=\small, rectangle),
  root/.style = (basic, rounded corners=2pt, thin, align=center,
    fill=blue!30, text width=4cm),
  level 2/.style = (basic, rounded corners=6pt, thin, align=center, fill=green!60,
    font=\footnotesize, text width=2cm),
  level 3/.style = (basic, thin, align=left, fill=pink!60, font=\footnotesize, text width=2cm),
  level 4/.style = (basic, rounded corners=6pt, thin, align=center, fill=red!40),
  level 5/.style = (basic, rounded corners=6pt, thin, align=center, fill=yellow!40),
}
% Set headers
\title[ME7244]{Foundations of computational materials modelling - Molecular dynamics simulation}
\institute[ITM]{Indian Institute of Technology Madras - Chennai 600 036}
\author[Narasimhan Swaminathan]{Narasimhan Swaminathan \inst{1}}
\institute[ITM]{\inst{1} Department of Mechanical Engineering, Indian Institute of Technology, Ma}
%=====
\begin{document}
\begin{frame}{}
\end{frame}
\end{document}
```



The image shows the same man from the previous slide, now looking at a different screen. The screen displays a visualization of a simulation box, likely a molecular dynamics simulation. The visualization shows a 3D representation of a simulation box with particles represented as small spheres. The box is shown from multiple perspectives (top, bottom, left, right). The software interface includes a toolbar with various icons and a panel on the right with settings and information. The man is gesturing towards the screen with his right hand.





Now, what happens if you actually pull it even more? So how do we actually, I am going to exercise 15 here, I will tell you what it does. Let us look at this, so the idea is to have a long nano size or not nano some, some material with free surface. So, this is the same copper material, except that now I have an outer simulation box and an inner region in which I am actually filling the atoms.

So, I am not filling the entire region with atoms but something else. So, simulation box is the outer part and then there is another region within it, into which the atoms are being filled. And now if I, if you keep looking at it, it is being pulled. And it is not just being pulled, after every bit of pull, it is being, energy is being minimized, the energy is being minimized because we are continuously pull it, the pull rate is so fast that the atoms are not going to occupy their minimum energy positions unless I actually minimize it for every step of

pulling. So, you see that when you look at the deformation at some particular point of time, it is, it is okay, but then all of a sudden you have the shearing shear bands, and all those things will start appearing.

So, in order to look at this a little bit more clearly, so what is basically happening is that you are, the slip. You have studied in material science. That is going to be slip. So, you can see those slip bands somewhat. I will I will take a look at this input file in a second, but if you do something called as the common neighbour analysis, so, common neighbour analysis is a technique which is capable of identifying atoms belonging to a FCC or a HCP or a BCC or icosahedral Lattice.

So, if you just say, perform common neighbour analysis, it is basically going to colour the atoms depending upon whether they are a part of the FCC, HCP BCC, icosahedral or other, something else. So, if you take a look at this, everything else that is not in the centre is marked as other because they are atoms belonging to the free surface. So, they will not be FCC or BCC, they will relax in a slightly different way.

So, if you continue to keep looking at the colours, all of a sudden you start, some BCC atoms begin to appear and then you have some failure. So you can actually look at this, a little bit better if you do select type, select the other atoms, that is the outer shell atoms and basically delete them. So, where is this? Delete selected. And then look at only what is happening to the FCC structure. So, initially everything inside is FCC for a particular value of strain. This is again a very simple calculation, very simple small input script and you have the deformation. You have all sorts of shear bands and other things happening.

So, you see that 45 degree plane here that is over which it is actually slipping, so you can see whatever you study in your book, you can actually see it happen when you actually simulate these simple things with little apps. You can do a better job. I need to play with the input file a little bit more. But let me show you what the input file looks like. Because this time, it is just not enough if you just continuously pull it, because you have to allow these atoms as you, previously we just pulled it a little bit, so it did not really matter, we were within the elastic regime. But now we are going beyond that. So, for every pull, you might want to make sure that the atoms are reaching their equilibrium positions for that stretch. Only then we will you be able to capture this phenomenon. If you continuously pull you do not see that.

(Refer Slide Time: 34:32)

```
13 #Lattice units is used, which is the default
14 #in the argon example, we used box units.
15 create_box 1 myreg
16 create_atoms 1 region fillreg
17 # Minimize using Embedded atom method potential for Al
18
19 pair_style eam
20 pair_coeff * * Al_jmp.eam
21 minimize 1e-25 1e-25 10000 100000
22 #=====
23 #Total time is 100ps
24 #Total strain is 0.02
25 #Strain rate is 0.02 per ps, which is 0.002x1012 strains per second
26 #2x109 strains per second!! Very high
27 #=====
28 reset_timestep 0
29 timestep 0.005
30 variable lz equal lz
31 variable newlz equal ${lz}
32 variable deltz equal 15.5840/79.72
33 dump_dmpid all custom 100 deformtensile.lammpstrj id x y z
34 label repeat
35 variable inte index 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
36 variable ndeltz equal ${deltz}*${inte}
37 compute str all pressure NULL virial
38 print 'DELTA z = ${ndeltz}' append DELTA.txt
39 fix defz all deform 1 z delta 0.0 ${ndeltz} units box
40 thermo_style custom c_str(3)
41 run 1000
42 unfix defz
43 minimize 1e-25 1e-25 100 1000
44 variable clz equal lz
45 variable szz equal c_str(3)
46 variable ezz equal (${clz}-${newlz})/${newlz}
47 print '${ezz} ${szz}' append strain.txt
48 uncompute str
49 next inte
50 jump Al.in repeat
51
52
```



So so again, I started off with start off with a sort of FCC structure and everything, so all the same, everything is pretty much same. But what I do here is I have introduced a couple of new things here, what do I want to do, I want to be able to apply some strain to the top face, minimize the energy, apply a little bit more, minimize the energy, apply a little bit more, minimize the energy until some finite strain is reached. So, right now I am applying a total strain of 15.5840 divided by 79.72, whatever that is, I think that is a 20 percent strain. And I am saying that the amount by which I want to extend my delta are of my simulation box in the z direction is 15.580 divided by 20. So, I define a new variable which is this delta z times 1.

So, initially it is it is 15.5840 divided by 20 times 1 and I calculate a new variable n delta z and apply that to my fix all deform command. This label repeat is actually a label in the input script, so that I can use this jump command to go back here and re-repeat it for the next one and it will add it to the to the actual value that it had reached previously. So, if if lz was actually the first length and in the first step it did lz plus delta z. The second step when it comes it has to do lz plus 2 delta z to lz plus 3 delta z. So, there is a reason all this is 1 because I always have to add the same amount to the final length. So, this could be actually modified and made a little bit simpler obviously.

So, and I, I try to achieve each deformation. So, each delta z is achieved in about 1000 steps. Following that, I minimize the command and I make sure that I unfix it because next time when it go, when it when it goes back to the same, when it goes back to the loop it will find a

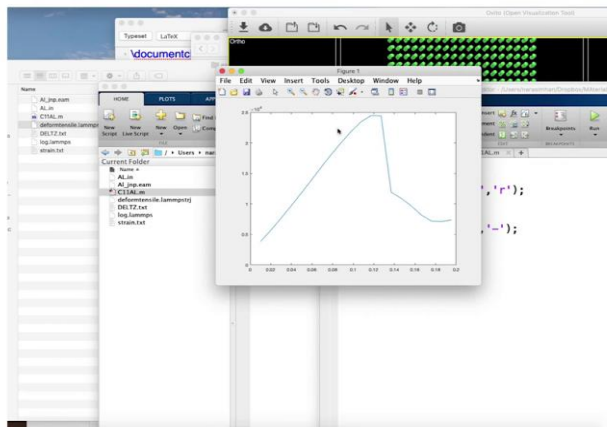
fix which was already defined unless I unfix it, it is going to give me an error. So, I unfix the deformation. So, I do this achieve a small delta z in 1000 steps minimize the energy, minimize the energy, minimize the energy for about 20 steps, in about 20 steps, I will reach 15.5840 total delta z and if you look at the I have to see if my strain (calculation) and then, sorry simultaneously I print the strain and the stress and append it to a text called strain dot txt. So, as is when it is calculating the stress and the strain it is going to fill it up with this line.

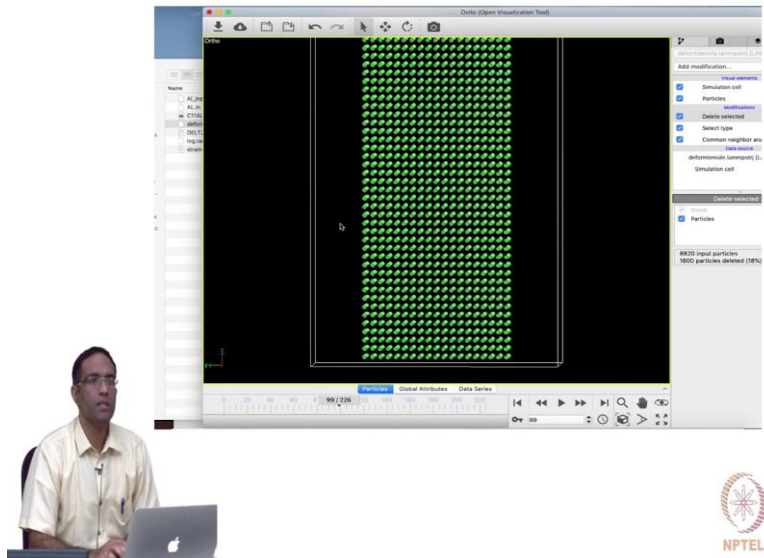
(Refer Slide Time: 37:46)

```

2 0.8195484194681386 -5625.25984745829
3 0.8293226292822077 -7546.77726885682
4 0.8398968389362769 -9558.84915611872
5 0.8488718466783462 -11639.2322865367
6 0.8586452584844155 -13749.7434678386
7 0.8684194681384848 -15982.9651721625
8 0.8781936778725539 -18882.12778624
9 0.8879678876866232 -19968.2399298616
10 0.8977428973486924 -21828.5488673875
11 0.18716387874762 -23458.978449435
12 0.11729851688831 -24472.990889518
13 0.1278667265429 -24432.2884692217
14 0.136838936276969 -11921.829322625
15 0.146613146811839 -18926.8871665864
16 0.156387355745188 -9968.7663464874
17 0.166161565479177 -8184.79444981317
18 0.175935775213248 -7194.87664117784
19 0.185709989472127 -7189.29825327586
20 0.195484194681386 -7398.81788756181

```





So, let us see whether this, yeah so, it contains about 20 steps about 20 steps and if we actually plot it you will see some non-linearity occurring obviously. So, it is just going up, is failing and just fail here, something has happened at this point. Until about in this case until about 10 percent strain, you are not seeing much of yeilding, which is, is that usual? No, it is not usual. You are looking at, this happening because it is not the bulk material number one, number two, it is a single crystal copper, single crystals are extremely strong, they have very high and then you are also pulling it at a very high rate no matter what you do. So, all these things are contributing to this sort of a behaviour, but you can see some failure actually happened when you brought the stress was a strain diagram.

So, these are some simple techniques that you can use in order to study the elastic properties of the materials. Now, these are extremely crude ways. Now you can, there are more elegant ways by which you can actually define the elastic constants itself as a function of the velocities and positions of all the atoms in the system and calculate it using statistical mechanics. So, once you know that you have to print out the positions and the velocities and then explicitly use that expression to evaluate your elastic constants of the material. So, anything else? Any other questions?

Professor: The shear or the normal?

Student: Normal. where you are printing a...

Professor: Yeah yeah.

Student: Those are Lattice constants; those are elastic constant.

Professor: Which? No, they are the stresses. So, this so this is a compute right.

Student: first we are ...

Professor: This is thermo style, custom means whatever I want to print I will print. Step is the step number, time is a time it is that is taken. C underscore str1 is taken from.

Student: Those are normal stresses, first 3.

Professor: First 3 are normal stresses, this next 3 are shear stresses and then this is the strain that I calculate here.

(Refer Slide Time: 41:15)

```
3 boundary      p p p
4
5 #fcc unit cell
6 variable a index 3.986
7 lattice fcc $a)
8 region        myreg block      -30 30 &
9               -30 30 &
10              -39.86 39.86 units box
11
12 region        fillreg block -19.93 19.93 -19.93 19.93 -39.86 39.86 units box
13 #Lattice units is used, which is the default
14 #in the argon example, we used box units.
15 create_box    1 myreg
16 create_atoms  1 region fillreg
17 # Minimize using Embedded atom method potential for Al
18
19 pair_style     eam
20 pair_coeff     * * Al_fcc.eam
21 minimize 1e-25 1e-25 100000 100000
22 #=====
23 #Total time is 10ps
24 #Total strain is 0.02
25 #Strain rate is 0.02 per ps, which is 0.002x1012 strains per second
26 #2x109 strains per second!! Very high
27 #=====
28 reset_timestep 0
29 timestep 0.001
30 variable il equal lz
31 variable newil equal $(il)
32 variable deltz equal 15.5846/20
33 dump dumpid all custom 100 deformtensile.lammpsstrj id x y z
34 label repeat
35 variable inte index 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
36 variable ndeltz equal $(deltz)*$(inte)
37 compute str all pressure NULL virial
38 print 'DELTA z = $(ndeltz)' append DELTA.txt
39 fix defz all deform 1 z delta 0.0 $(ndeltz) units box
40 thermo_style custom c_str[3]
41 run 1000
42 unfix defz
43 minimize 1e-25 1e-25 100 1000
44 variable clz equal lz
45 variable szz equal c_str[3]
```



The instructor clarifies how to calculate the strain



```

10                                     -39.86 39.86 units box
11
12 region fillreg block -19.93 19.93 -19.93 19.93 -39.86 39.86 units box
13 #lattice units is used, which is the default
14 #in the argon example, we used box units.
15 create_box 1 myreg
16 create_atoms 1 region fillreg
17 # Minimize using Embedded atom method potential for Al
18
19 pair_style eam
20 pair_coeff * * Al_jnp.eam
21 minimize 1e-25 1e-25 100000 100000
22 #
23 #Total time is 10ps
24 #Total strain is 0.02
25 #Strain rate is 0.02 per ps, which is 0.002x10^(12) strains per second
26 #2x10^9 strains per second!! Very high
27 #=====
28 reset_timestep 0
29 timestep 0.001
30 variable il equal lz
31 variable newil equal $(il)
32 variable deltz equal 15.5846/70
33 dump dumpid all custom 100 deformtensile.lammpstrj id x y z
34 label repeat
35 variable inte index 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
36 variable ndelz equal $(deltz)*$(inte)
37 compute str all pressure NULL virial
38 print "DELTA z is $(ndelz)" append DELTA.txt
39 fix defz all deform 1 z delta 0.0 $(ndelz) units box
40 thermo_style custom c_str[3]
41 run 1000
42
43 unfix defz
44 minimize 1e-25 1e-25 100 1000
45 variable clz equal lz
46 variable szz equal c_str[3]
47 variable ezz equal ($(clz)-$(newil))/$(newil)
48 print "Ezz[3] $(szz)" append strain.txt
49 uncompute str
50 next inte
51 jump AL.in repeat
52

```



So, in this in the next exercise, in in exercise 15, the strain is calculated in a in a careful way. So, what is happening is/, see you have to think about it a little bit here. So, when it comes to repeat it is going to, this command is going to add the delta z value to the already expanded system, so to the already expanded lz only it is going to add. But the, in the calculation of strain the value that we want to subtract, you know the final length minus the initial length divided by the initial length, this is supposed to be the original 72 point whatever, the, the actual thing that we started off with basically twice 39.86.

So, you have to store that value here. So, il equal lz and new il equal to dollar il. So, when you do this, the new il value does not change. It is fixed to the value that was initialized. So, when you use the dollar new il value, you will be subtracting it from 72 point something, something and here also 72 point something, something and this will be the new length and you will get the corresponding epsilon zz.

This is something that you that I had to work with for some time to actually figure out. So sometimes you have to do some simple tests and make sure that you understand what whatever this gives. Yes?

Student: Sir, we are having such high strain rate?

Professor: Yeah.

Student: Like, minimization will it happen actually like in such a high strain rate...

Professor: No, you are, I am actually applying a strain to deform it to a certain extent. And then minimizing the energy of the system, during the process of tuning usually not much happens because it is quite, because it is, I am not giving any time for the atoms to actually move.

Student: So...

Professor: In the...

Student: Sir, will not the simulation be close to actual when we do not have the minimize, the thing during the preparations, because it seem like it is so high that.

Professor: It is so high. So, in order, if you, I do not I do not understand your question. So, you are telling me, if I did not have the minimization, what would happen?

Student: Like, since the strain rate is so high, minimization will not happen in the actual system.

Professor: Actual means in a real?

Student: Like in the real case, the atom does not have time enough to minimize their value.

Professor: Right, right. So, this is not at that strain rate. This is not at that high strain rate. The total experiment is actually not on an average at that strain rate, because I have a allowed some minimization. If you want the total experiment to be at that strain rate, I should not minimize, I just pull.

Student: But if the simulation itself is having such high strain rate, right?

Professor: This portion, this portion, the small portion, where Δz is added is at a high strain rate, but then I stop it and minimize it. So that basically negates hopefully a little bit of that strain rate, high strain rate effect. The, the average experiment, the computational experiment that we have seen now, the overall experiment is not conducted at a high, at the high strain rate that you would get if you divide the strain amount by the time because you are minimizing in between. However, this is a little bit more realistic of the real system than if you continuously pulled at such high standards because you cannot perform experiments at such high strain rates. So, normally what we do is we actually pull a little bit, minimize it,

pull a little bit minimize it. So, even though the elastic constant determination is the methodology is very clear, the actual simulation can take quite a long time.

Student: Sir, why do...

Student: Sir, where are we defining 2 different region in the script like how outer?

Professor: Good, that is a good question, I think I should have highlighted it. So here, there is a region, I say my region block, and I am giving some arbitrary values minus 30 to 30, minus 30 to 30 in the x and the y directions, and in the z directions it is exactly equal to 20 times the lattice constant. And then I will fill another region and I call it filled region because that is a region I am going to fill, which is smaller than the outer region. But the z direction exactly spans my box. Now, when I say create the box, it creates with my region, when I say create atoms, I fill it inside this. So, it just fills it exactly inside this. So now, this is very nice because when you want to simulate, say a sphere of atoms or put atoms in a cylinder, then you can always define this additional region and fill atoms inside that region.

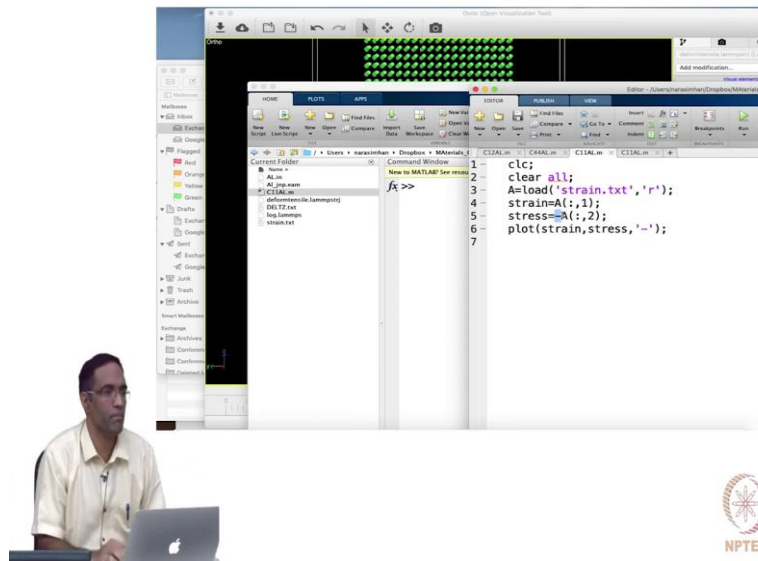
Student: What would a temperature, the initial temperature of it, does it minimize considering that, here we are not defining anywhere temperature.

Professor: We have not defined, this is just a pure molecular kind of static.

Student: So what, what is the temperature?

Professor: 0 K. Yeah, the atoms are not even vibrating about their mean positions in this case. But if you want more realistic elastic constants to come out of your simulation, you need to include those effects, because the, the expression for elastic constants is not only dependent on positions, but also on the velocities. We have just removed that completely because we just wave our hands and say that for solids it is not that important.

(Refer Slide Time: 47:17)



Student: Sir, when you said it is calculating the pressure tensor, pressure is designed to be the (isotropic) isotropic part of the stress tensor.

Professor: No, no, no, it is not that, it is not it is not the hydrostatic part of the, it is not hydrostatic stress. It is, it is stress except there it is having negative, so that is why that is why I added a minus sign in your, in my stress here to make it appear positive because all negative stresses means tensile, positive stresses means, yeah, sorry, yeah negative stresses in in LAMMPS, when it is printing out, it is actually printing out the tensile stresses, this the stress that is acting in the other direction is, is negative.

Student: Pressure it is just calculating sigma.

(Refer Slide Time: 47:57)

12. Use Python with LAMMPS
13. Errors
14. Building the LAMMPS manual

Examples

```
compute 1 mobilid stress/atom Mobil  
compute 2 mobilid stress/atom mobilp  
compute 3 all stress/atom NALL pair bond
```


Description

Define a computation that computes the symmetric per-atom stress tensor for each atom in a group. The tensor for each atom has 6 components and is stored as a 6-element vector in the following order: xx, yy, zz, xy, xz, yz. See the compute pressure command if you want the stress tensor (pressure) of the entire system.

The stress tensor for atom I is given by the following formula, where a and b take on values x,y,z to generate the 6 components of the symmetric tensor:

$$S_{ab} = - \left[m v_a v_b + \frac{1}{2} \sum_{n=1}^{N_n} (r_{1n} F_{1n} + r_{2n} F_{2n}) + \frac{1}{2} \sum_{n=1}^{N_n} (r_{1n} F_{1n} + r_{2n} F_{2n}) + \frac{1}{3} \sum_{n=1}^{N_n} (r_{1n} F_{1n} + r_{2n} F_{2n} + r_{3n} F_{3n}) + \frac{1}{4} \sum_{n=1}^{N_n} (r_{1n} F_{1n} + r_{2n} F_{2n} + r_{3n} F_{3n} + r_{4n} F_{4n}) + \frac{1}{4} \sum_{n=1}^{N_n} (r_{1n} F_{1n} + r_{2n} F_{2n} + r_{3n} F_{3n} + r_{4n} F_{4n}) + \text{Kspace}(r_{1n}, F_{1n}) + \sum_{n=1}^{N_n} r_{1n} F_{1n} \right]$$

The first term is a kinetic energy contribution for atom I. See details below on how the specified temp-ID can affect the velocities used in this calculation. The second term is a pairwise energy contribution where n loops over the Np neighbors of atom I, r1 and r2 are the positions of the 2 atoms in the pairwise interaction, and F1 and F2 are the forces on the 2 atoms resulting from the pairwise interaction. The third term is a bond contribution of similar form for the Nb bonds which atom I is part of. There are similar terms for the Na angle, Nd dihedral, and Ni improper



Student: Sir, there is also a compute stress in manual?

Professor: That is compute stress per atom. We do not want that now. Per atom stress, now what that means, I do not know but because you can think about it, what is meant by stress per atom? You still you still can get useful information, for example, if you model a (dislocation), if model a grain boundary, and if you model the stress per atom and average it nicely, you will see the dislocation course alone stress, you must have studied that in some course.

You have stresses emanating from the dislocation core, going reducing as 1 by r square or something like that. So, you can look at all that. So, this is actually the stress that is counted per atom. But that is not what we want, we do not want per atom stress information. We want the stress for the entire simulation boxes and so global quantity, so compute pressure is what is going to do that.