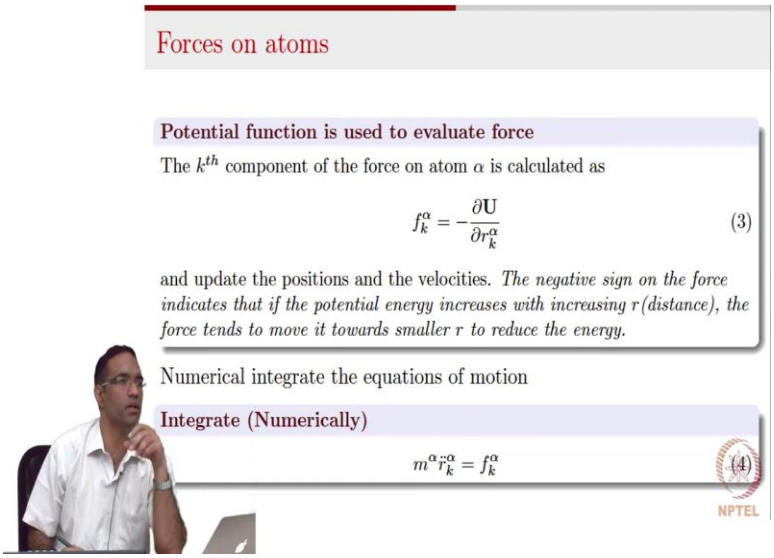
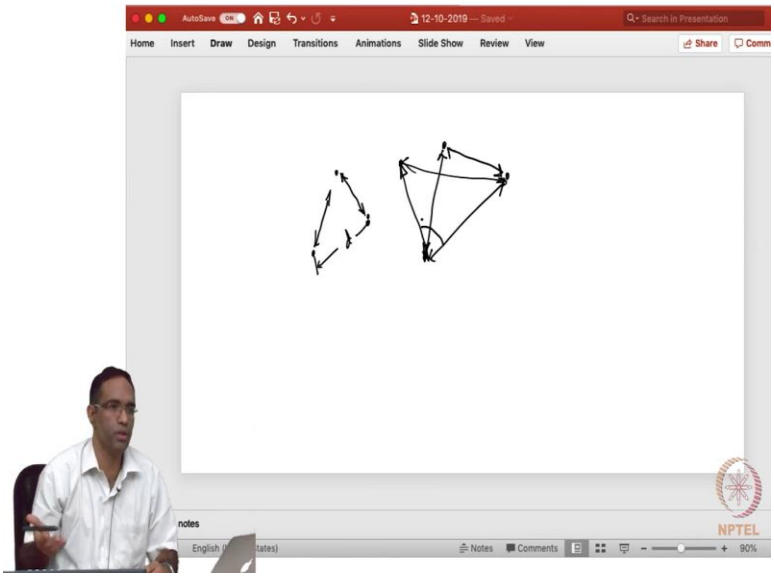
Foundations of Computational Materials Modelling
Professor Narasimhan Swaminathan
Department of Mechanical Engineering
Indian Institute of Technology, Madras
Input script for LAMMPS 1

(Refer Slide Time: 0:16)



The first one is the presence of this negative sign in calculating the forces. So, though we do this, so routinely that we generally forget us to why that is being done. So, it should take a spring system two masses, which are connected by spring, then if you increase the distance between them, the potential energy of the system increases. So, the negative sign is basically to indicate that, the force is going to act in a direction, so as to decrease the distance in order to decrease the potential energy of the system. So, that is the reason why you have the negative sign in that particular location there. The next question is regarding gravitational potential energy. So, we only consider pairwise interactions, simply because we think that the three body interactions is actually weak. So, while saying that, it is important for me to mention the following.

(Refer Slide Time: 1:12)



So, if you have two atoms and the equilibrium distance of separation between them is d, the presence of the third one even, though you consider only pairwise interactions between them is actually going to change this d. It is not going to remain constant, okay ? Three body interactions or those interactions which alter the behavior of the system in over the two body interactions, anything that is affecting it beyond the two body interactions or pair interactions is the three body, anything that is affecting the system's behavior beyond the three body interaction is the four body interaction. So, we really do not know, we really have to see, the actual behavior of the system in order to examine whether a three body interaction is required or not. So, with these two clarifications I think we will begin talking about LAMMPS. I will start talking about the input file and various commands, that are used there.

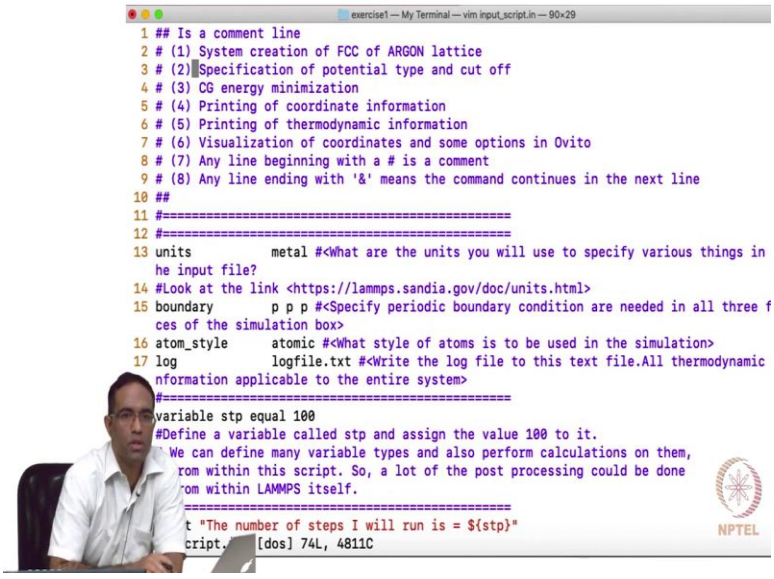
Student: Examples of three body potentials and why they are used?

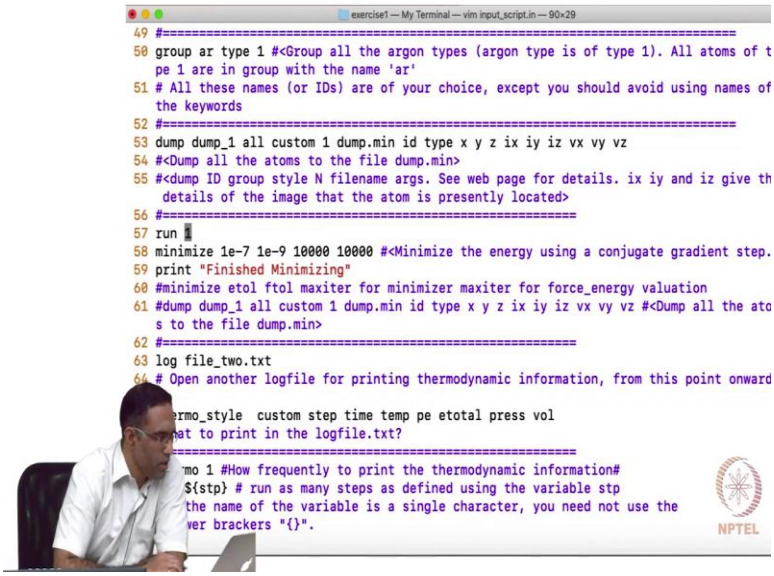
Professor: For example, if you talk about, say let us talk about I do not know whether this is the right example to give, right now the interaction or the energy between the energy of the system only depends upon this distance, and these distances. Now, imagine this atom being actually present at exactly the same distance somewhere here. Now, the question is yeah these distances actually affect the behavior of the system, but thus this angle actually effect the behavior of the system.

So, in case of Silicon Carbide, the bonding is such that, you have to have the tetrahedron formed, you must have say there is specific units set are formed in silicon diamond structure, or this tetrahedral structures that are formed. So, apparently the angles between the various bonds are also extremely important, and those sort of that sort of behavior is actually captured through these three body interactions, therefore toss off potential and potentials like Stillinger-Weber potentials are all potential, which will have this angle term in some may, in some manner and they are therefore three body interactions.

We will come to that, I will discuss those potentials and a little bit more detail, when we discuss when we come to that point of the course, right now we will restrict our search to very simple potentials, because a goal is now first to get you started with writing some sort of a input script. And then see what sort of information you can actually gather from LAMMPS. And then we can slightly go advanced to ceramics, and probably metals in order to see how you specify potentials for these material systems.

(Refer Slide Time: 4:13)





So, so, before I begin the input script for LAMMPS, there are certain rules and regulations that you have obviously have to follow when you are writing the input script. So, anything that begins with the hash is a comment line. So, it is a good idea to actually always comment your input script and tell what is what a particular line is actually going to. For yourself and for somebody else who is actually going to use your script at a later stage, writing an input script without commenting it is going to be disastrous for you and for somebody else who is going to carry over your work.

So, always have a hash and put something, something about what you are going to do in your input script. In this simple example, we are not going to do any complicated simulation, the basic idea is only to introduce some of the commands, simultaneously we will switch between this file that you are looking at here, which can be written in any text editor. And also the LAMMPS documentation, and I will show you different regions in the documentation, where you need to carefully look in order to find further information about a particular command.

Because I am only illustrating certain aspects of a particular keyword here. We will see how we specify a potential type and a cutoff for that particular potential. We are going to be looking at a noble system, noble gas system such as Argon in this example. So, we will use the Lennard jones potential, which I introduced in the last class. We will do some energy minimization, as I do not know whether I have a command for the energy minimization probably not on this input script, but I do have it here. So, I will talk about that a bit.

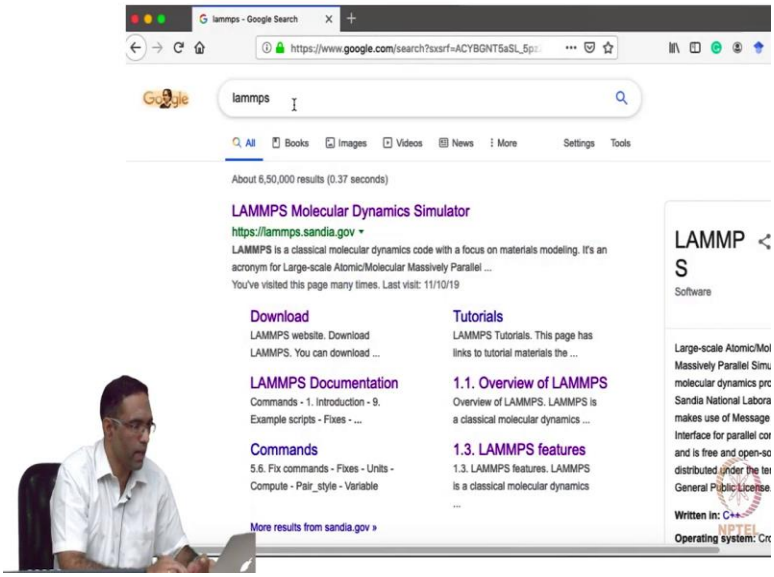
And then how do you print out the information of the system like the X, Y, Z coordinate. So, as I mentioned the whole goal of this LAMMPS is to basically solve equations of motion. And print out the positions and the velocities of all the atoms. So, you need these positions and the velocities of the atoms in order to post process other information. So, you need to print them out for visualization purposes, or for getting other information, of any thermodynamic property, which may be depending on the position and the velocities of all these atoms. The printing of thermodynamic information.

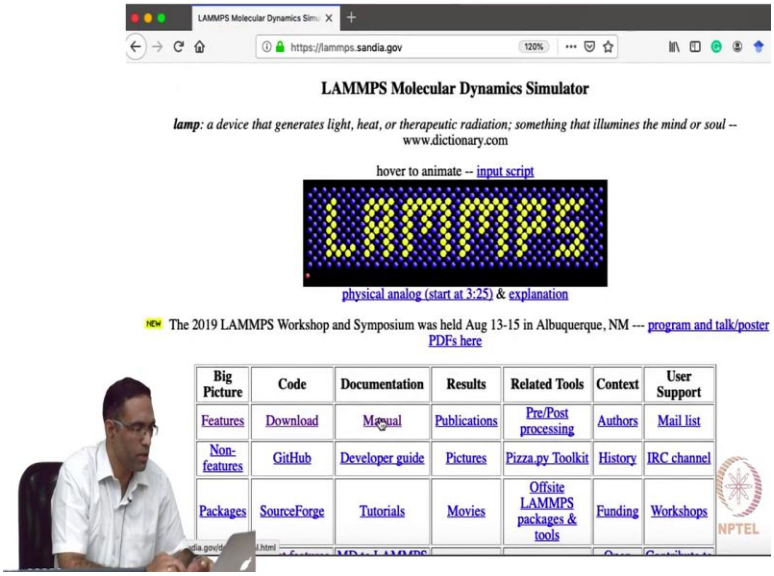
So, like I mentioned there are two kinds of things that you can print out, one is one concerns the entire simulation box, for example the pressure of the entire system, or the volume of the entire system, or the temperature of the entire system and so on. And there are, there are other kinds of information, which are specific for each and every atom that constitute the system. So, per atom information needs to be printed out. So, the coordinate information will constitute, you know per atom kind of information, and thermodynamic information will be like the pressure, temperature, and all that. So, we need to print out these two useful things, visualization of the coordinates and some options that we can look at in Ovito, may be the options will come at a later stage. And then like I mentioned we need to comment it.

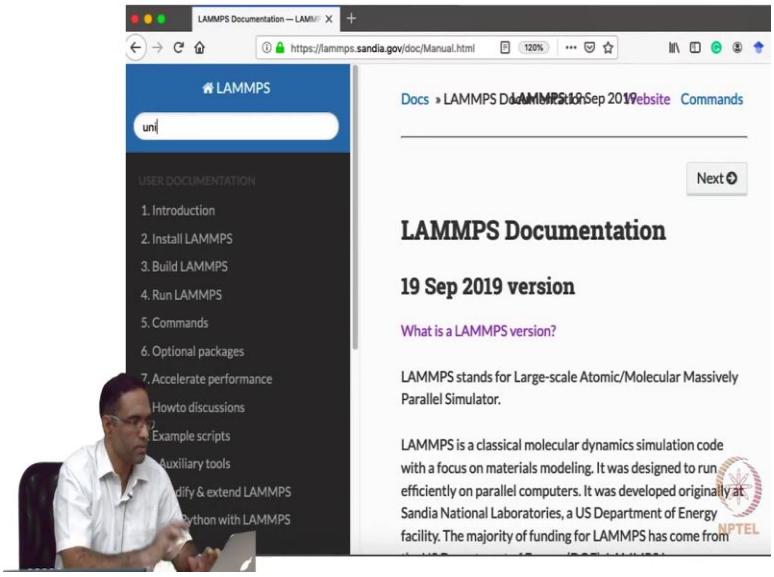
So, I just mentioned there any line beginning with the hash is a comment line, and any line ending with a ampersand means the command or the various inputs to that particular command are continuing in the subsequent lines. So, these are the things that we might want to remember. So, these may be available at different points, or different regions in the, in the manual itself. But I think it is a, it is a good thing to know these things first before, we start writing any input script.

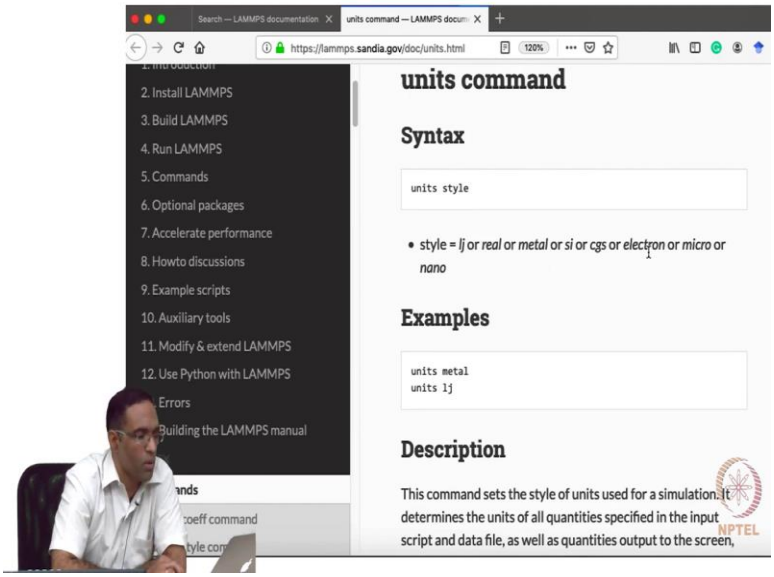
So, the first thing that you need to specify in the LAMMPS input script, it should tell what units you are going to use, for example if you say that the in LJ units, you specify epsilon and sigma. The energy as well as the distance, sigma has units of distance, and epsilon has units of energy. So, what energy units are going to specify, if at all you are going to specify epsilon, and a sigma, what units are they in currently? So, that is the first thing, that we use and I have used a units called metal units. So, let us see what other units are there in LAMMPS.

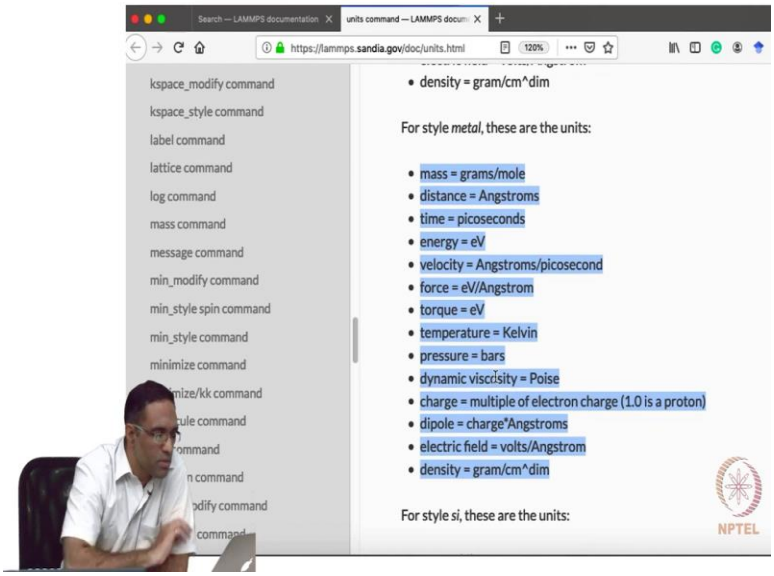
(Refer Slide Time: 8:40)






LAMMPS Documentation — LAMM
https://lammps.sandia.gov/doc/Manual.html
120%
LAMMPS
uni
USER DOCUMENTATION
1. Introduction
2. Install LAMMPS
3. Build LAMMPS
4. Run LAMMPS
5. Commands
6. Optional packages
7. Accelerate performance
Howto discussions
Example scripts
Auxiliary tools
dify & extend LAMMPS
ython with LAMMPS
Docs » LAMMPS Documentation » Sep 2019 version
Website
Commands
Next
LAMMPS Documentation
19 Sep 2019 version
What is a LAMMPS version?
LAMMPS stands for Large-scale Atomic/Molecular Massively Parallel Simulator.
LAMMPS is a classical molecular dynamics simulation code with a focus on materials modeling. It was designed to run efficiently on parallel computers. It was developed originally at Sandia National Laboratories, a US Department of Energy facility. The majority of funding for LAMMPS has come from
NPTEL


Search — LAMMPS documentation
units command — LAMMPS docum
https://lammps.sandia.gov/doc/units.html
120%
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
Errors
Building the LAMMPS manual
units command
Syntax
units style
style = lj or real or metal or si or cgs or electron or micro or nano
Examples
units metal
units lj
Description
This command sets the style of units used for a simulation. It determines the units of all quantities specified in the input script and data file, as well as quantities output to the screen,
NPTEL



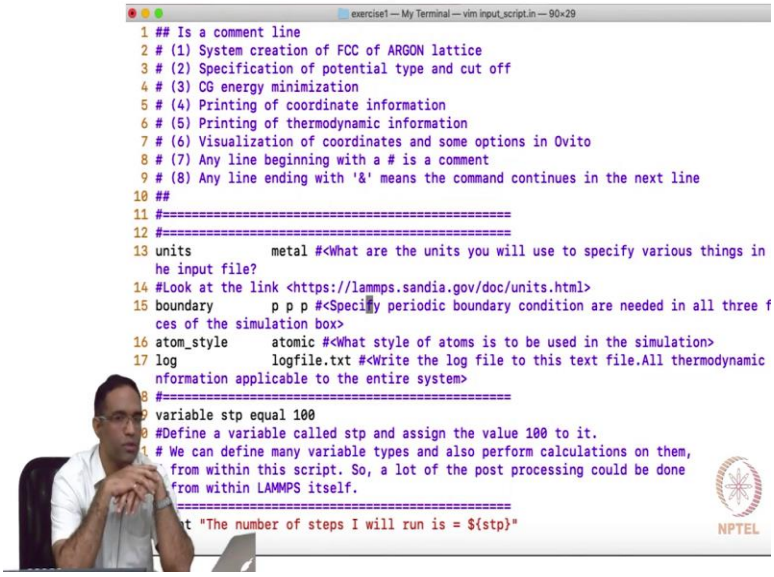
So, in order to do that I go to LAMMPS, I go to LAMMPS Molecular Dynamics Simulator the main web page, and I click manual and I see units. The first thing that turns up this is units command, and there is nothing, nothing very complicated about this the command says the keyword is units followed by something called style. And the style can be either LJ or real or metal or SI or CGS or electron or micro or nano. So, these are the various options available for the units, for the units.

And below you can find for each and every kind of units, what, what it means? So, for example if you look at style real, if you specify mass is 1, it essentially means 1 grams per mole, that is it. If you specify distance it is, if you specify some distance between two atoms is 1, it essentially means you are specifying the distance in Angstroms. So, in this manner by looking at these options you should actually know what units you have to specify for your problem, obviously when we are dealing with atomistic systems, we prefer the distance units to be in the order of angstroms the energy to be in the order of electron volts, temperature to be in the order of Kelvin and so on and so forth. So, the metal units is usually a convenient system to use.

So, if you have forces electron volt per angstrom. So, if you specify some where a force is something 1, 1, it is not 1 Newton, it is 1 electron volt per angstrom is what you are specifying. So, this is something that you need to keep in mind. The pressure that you are specifying or the pressure that is being outputted from the system will be in bars. So, if you want to convert this into mega Pascals or Giga Pascals in case you are talking about your mechanical properties of a

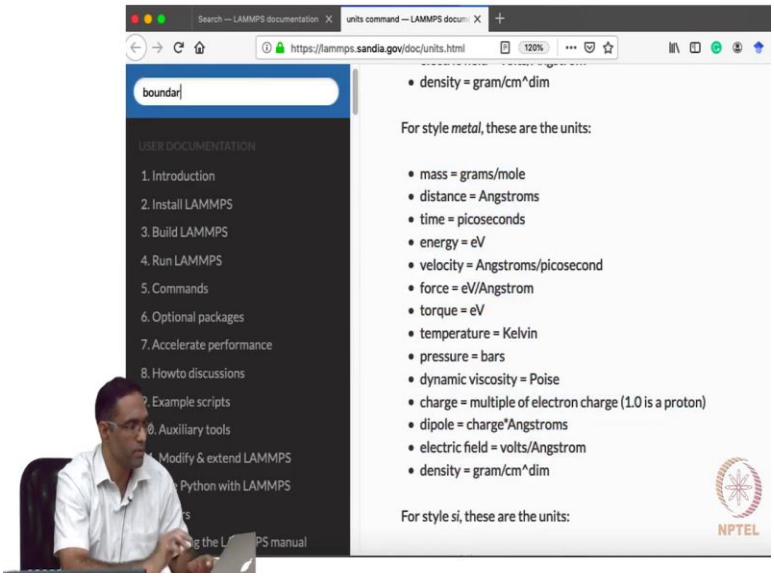
system, you need to perform the appropriate conversion. So, these kind of systems, now these kind of things you need to keep in mind while you are starting to write your input file.

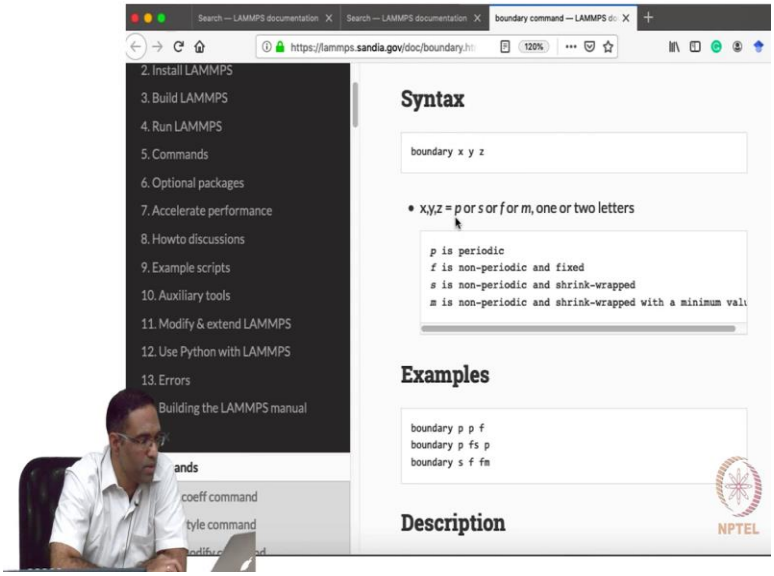
(Refer Slide Time: 10:53)

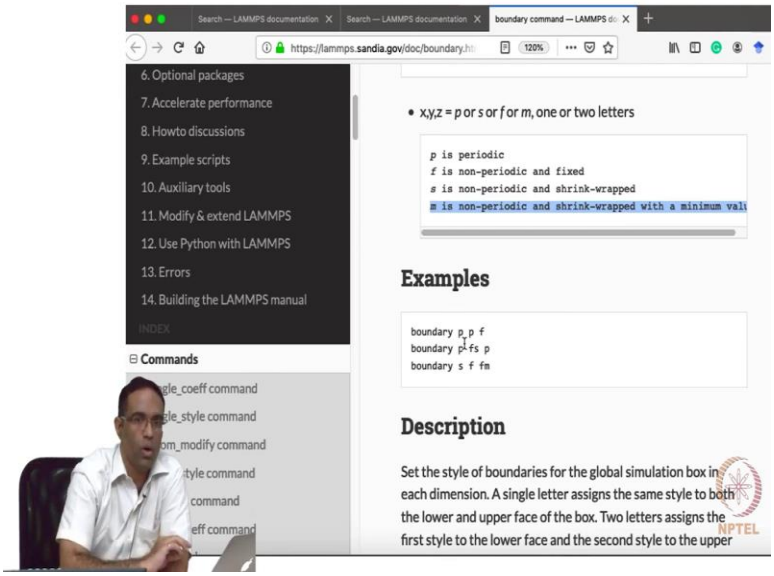


Boundary, so you need to specify whether you are going to use periodic boundary conditions, or not, like I mentioned yesterday periodic boundary conditions are essentially used to simulate the bulk behavior of the system, that means you do not want to have any free surfaces in your crystal. The behavior would be entirely different, if you had free surfaces, of course if you have a large enough the crystal. Then you the free surfaces play a very small role, but remember we are looking at systems comprising of maybe few thousand atoms or few ten thousand atoms. And if you are not going to simulate the right thing, then your, your answers can be you may be misinterpreting your answers, you may be simulating a system with free surfaces but you may want actually to simulate a bulk material. So, if you want to simulate the bulk material, you need to use p p p, where p p p refers to periodic boundary conditions on all three faces of the simulation box.

(Refer Slide Time: 11:56)

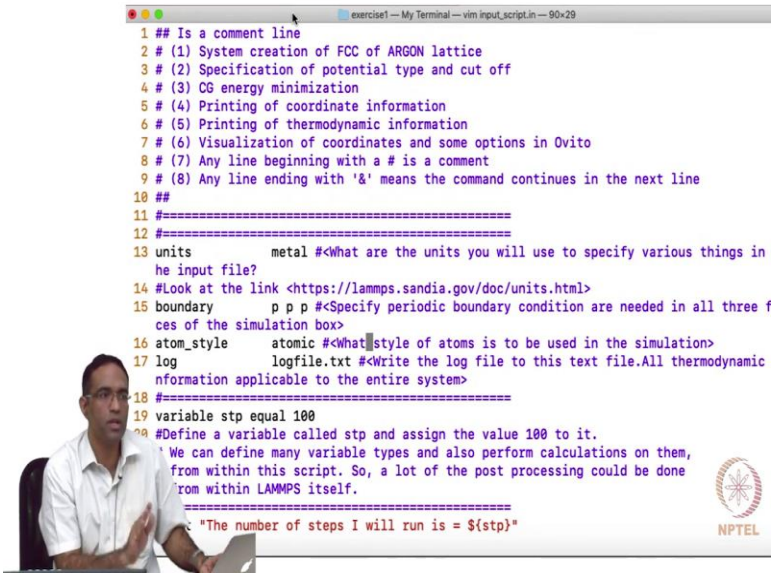


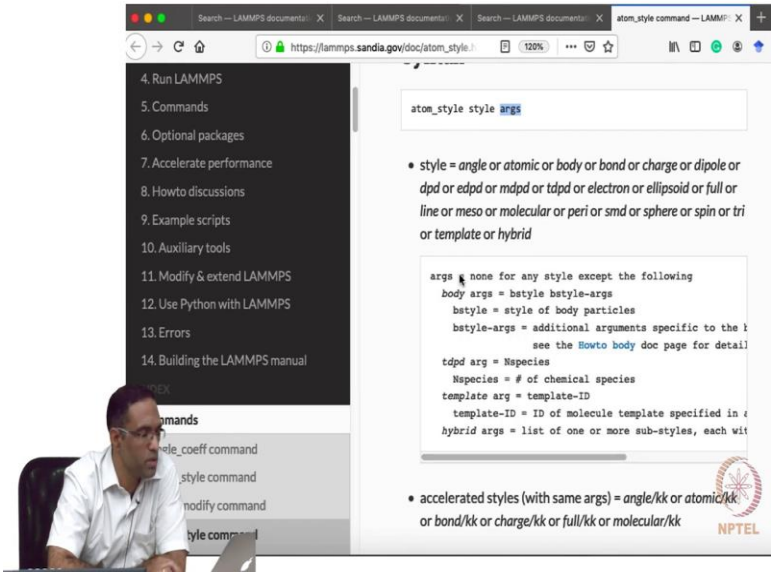


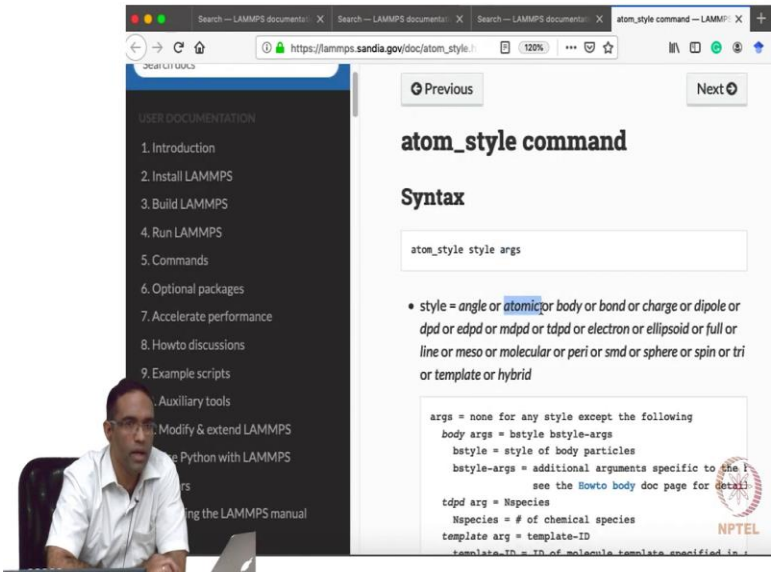


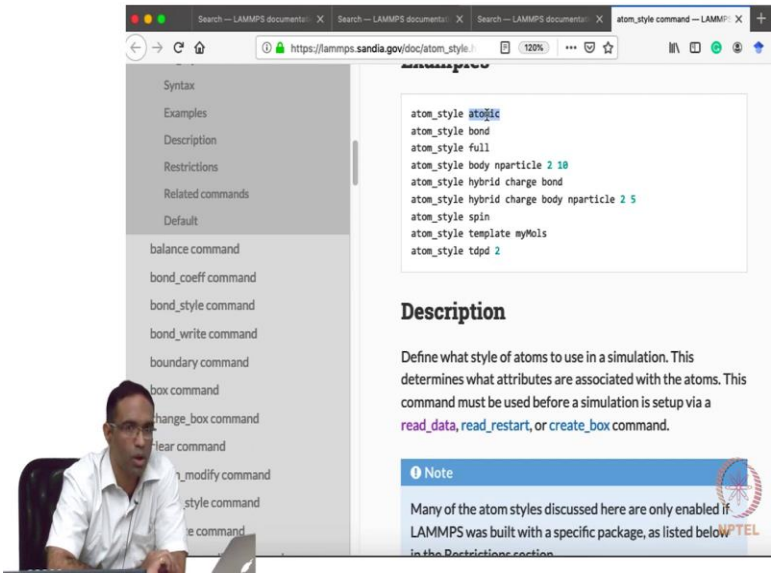
There are other options that are available in LAMMPS. And again I would just do this, so again boundary has X Y Z. So, X Y Z, p or s or f or m one or two letters. So, basically p is periodic, f is non periodic and fixed, s is a non- periodic and shrink-wrapped. So, we will come to we, we will see if this is required in some of the examples, usually in the examples that I am going to be discussing, we will only deal with periodic boundary conditions. So, we do not really need that, but if you need to know something about it you can actually go ahead and read the manual and it should be pretty straightforward. So, we will use periodic boundary conditions and obviously you have an option of specifying periodic for X conditions, and fixed for the Y faces, the faces with the, with the normals in the Y direction and so on. You have all these options and depending upon the problem that you are tackling you may have to change these things. For our case for our purposes, periodic is more than sufficient.

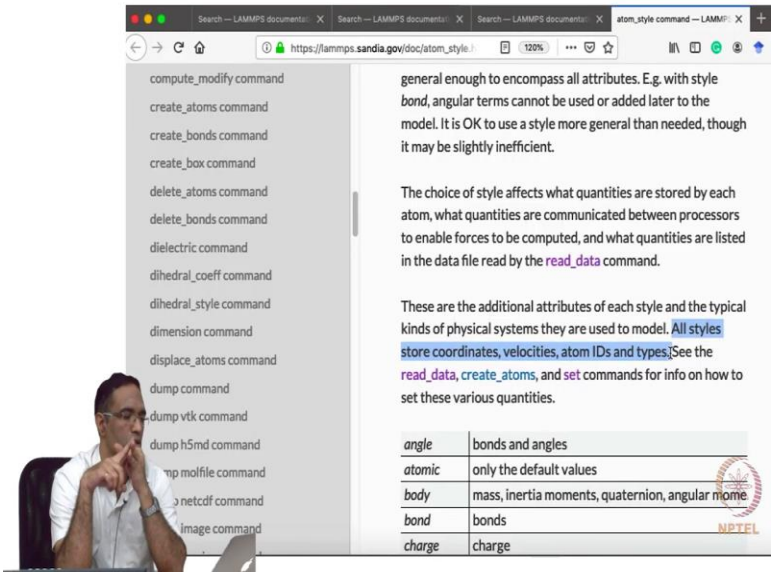
(Refer Slide Time: 13:08)






https://lammps.sandia.gov/doc/atom_style.h
USER DOCUMENTATION
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
Previous
Next
atom_style command
Syntax
atom_style style args
style = angle or atomic or body or bond or charge or dipole or dpd or edpd or mdpd or tdpd or electron or ellipsoid or full or line or meso or molecular or peri or smd or sphere or spin or tri or template or hybrid
args = none for any style except the following
body args = bstyle bstyle-args
bstyle = style of body particles
tdpd arg = Nspecies
Nspecies = # of chemical species
template arg = template-ID
NPTEL


https://lammps.sandia.gov/doc/atom_style.h
Syntax
Examples
Description
Restrictions
Related commands
Default
balance command
bond_coeff command
bond_style command
bond_write command
boundary command
atom_style atomic
atom_style bond
atom_style full
atom_style body nparticle 2 10
atom_style hybrid charge bond
atom_style hybrid charge body nparticle 2 5
atom_style spin
atom_style template myMols
atom_style tdpd 2
Description
Define what style of atoms to use in a simulation. This determines what attributes are associated with the atoms. This command must be used before a simulation is setup via a read_data, read_restart, or create_box command.
Note
Many of the atom styles discussed here are only enabled if LAMMPS was built with a specific package, as listed below
NPTEL



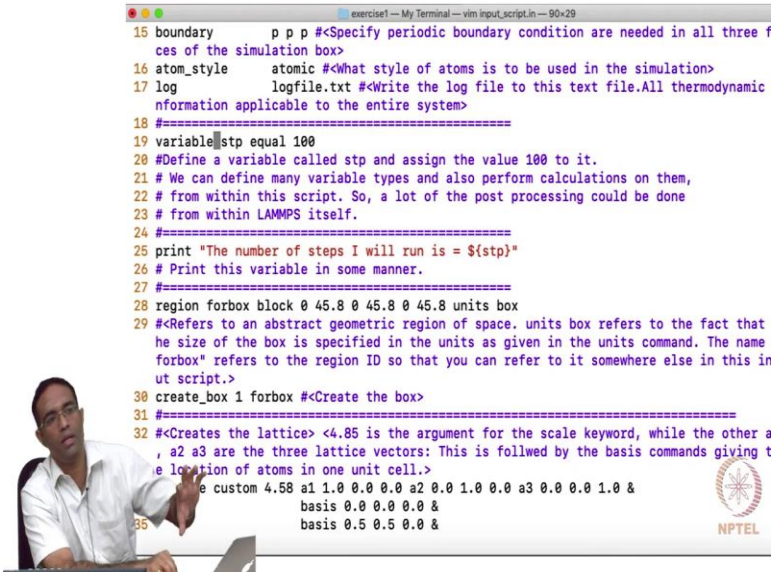
Then you have to specify atom style, atom style is specified as being atomic, these are basically what are your atoms. So, LAMMPS has an ability to actually simulate not just, see in our case what attributes does our atom have, our atom has the following, or it has a position in the velocity. But that may be systems that you may want to simulate, not related to molecular simulations of crystals and so on, where you can actually simulate spheres with a finite radius. For example, in a discrete element method, this is another meso scale method, where you have huge particles like for example if you want to simulate the crush behavior of sand particles, then each sand may be considered as a sphere with a finite radius. So, such things can also be done here with LAMMPS.

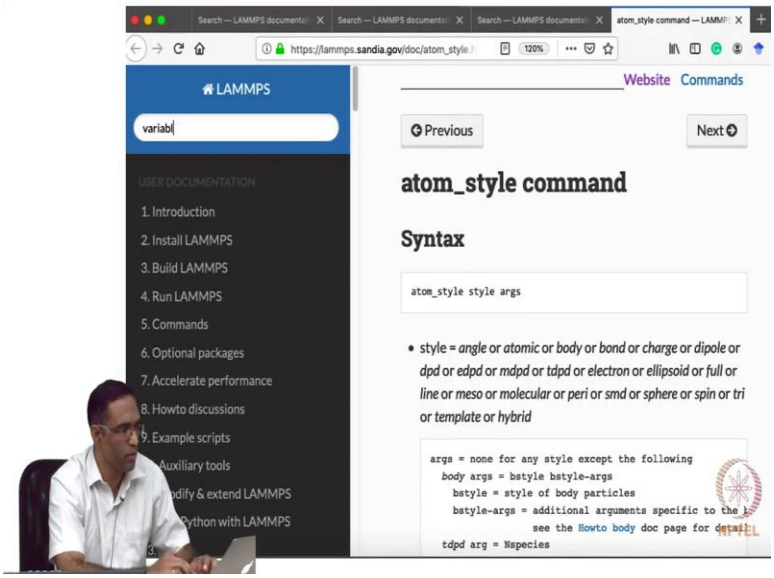
So, the style that we are interested in is the atomic style, because we are only interested in looking at the positions and the velocities of the atom and therefore the atom style will have the option atomic. So, there are so many style and the corresponding thing. So, so, far the commands that we saw was, it was having a key word and then it had a style. But now we are going to have atom style, we are going to have the key word, we are going to have a style and we are going to have corresponding arguments that we can give. So, it says below for example in this block here what are the arguments that is required?

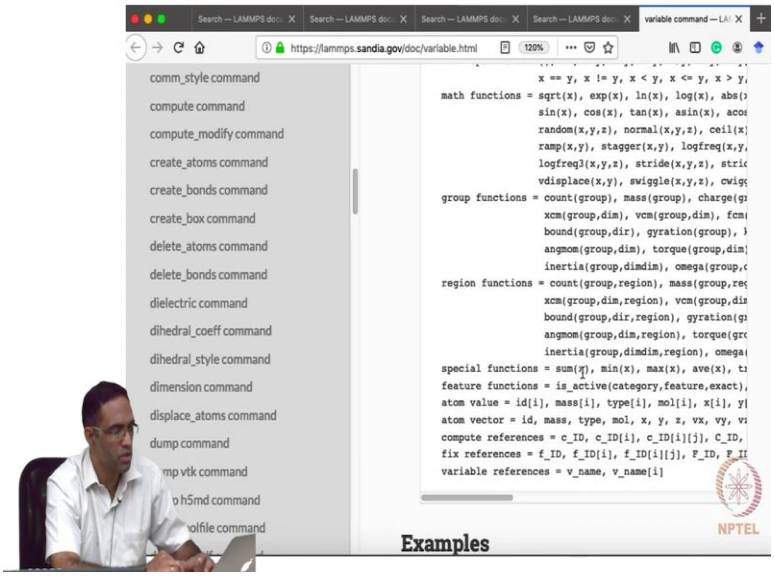
So, arguments is equal to none for any of the styles except for body, where the arguments are supposed to be bstyle and so on. But since we are using the atomic style we do not really have to give any arguments. If you are giving yeah, so you can go ahead and look at other styles if at all

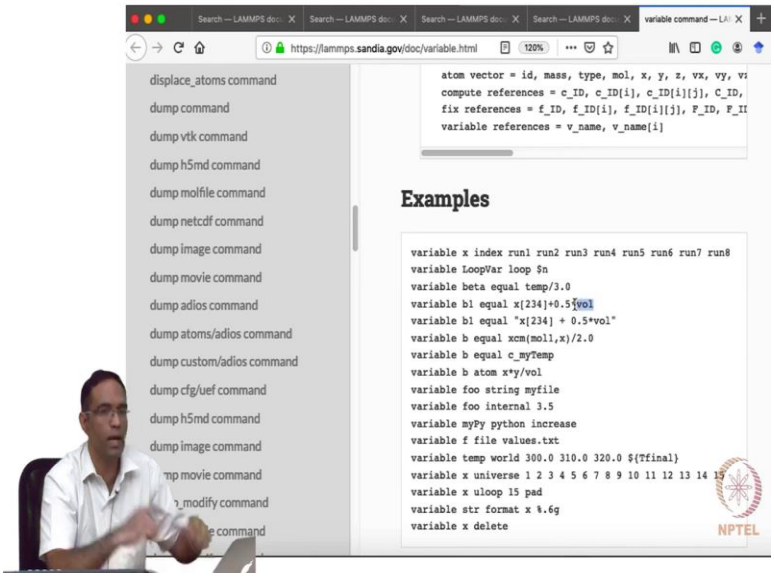
you want to, but for now we will be satisfied with the atomic style. And somewhere I saw so all styles will store coordinates, velocities, atom IDs and types. So, the atom style, when you say atom, you are going to be able to ask you can you can ask LAMMPS to know its positions, the velocities, the IDS of the atoms, and also the various types, you will specify the types in the future, in the input script, as I am just going to show you, charge type.

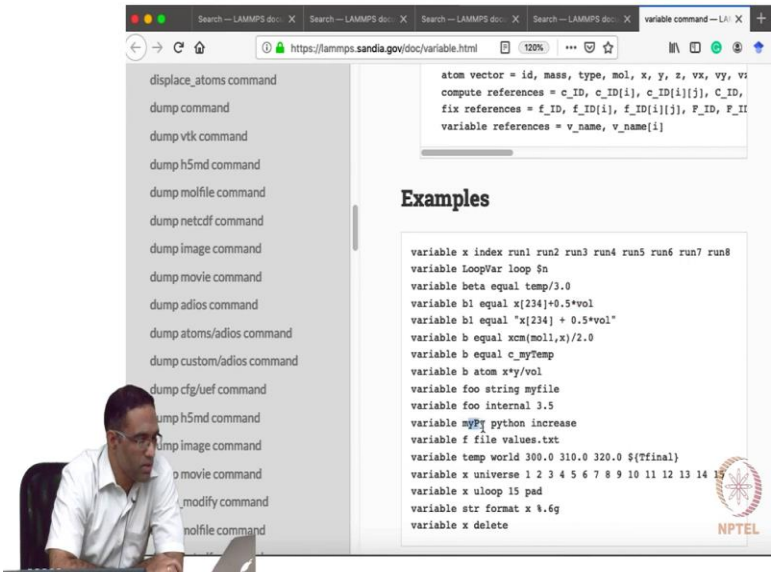
(Refer Slide Time: 15:40)






https://lammps.sandia.gov/doc/variable.html
comm_style command
compute command
compute_modify command
create_atoms command
create_bonds command
create_box command
delete_atoms command
delete_bonds command
dielectric command
dihedral_coeff command
dihedral_style command
dimension command
displace_atoms command
dump command
dump vtk command
dump h5md command
dump molfile command
math functions = sqrt(x), exp(x), ln(x), log(x), abs(x
sin(x), cos(x), tan(x), asin(x), acos
random(x,y,z), normal(x,y,z), ceil(x)
ramp(x,y), stagger(x,y), logfreq(x,y,
logfreq3(x,y,z), stride(x,y,z), stric
vdisplace(x,y), swiggle(x,y,z), cwigg
group functions = count(group), mass(group), charge(gr
xcm(group,dim), vcm(group,dim), fcm(
bound(group,dir), gyration(group),
angmom(group,dim), torque(group,dim)
inertia(group,dimdim), omega(group,
region functions = count(group,region), mass(group,reg
xcm(group,dim,region), vcm(group,dim
bound(group,dir,region), gyration(gr
angmom(group,dim,region), torque(gro
inertia(group,dimdim,region), omega(
special functions = sum(x), min(x), max(x), ave(x), tr
feature functions = is_active(category,feature,exact),
atom value = id[i], mass[i], type[i], mol[i], x[i], y[
atom vector = id, mass, type, mol, x, y, z, vx, vy, vz
compute references = c_ID, c_ID[i], c_ID[i][j], C_ID,
fix references = f_ID, f_ID[i], f_ID[i][j], F_ID, F_ID
variable references = v_name, v_name[i]
Examples
NPTEL


https://lammps.sandia.gov/doc/variable.html
displace_atoms command
dump command
dump vtk command
dump h5md command
dump molfile command
dump netcdf command
dump image command
dump movie command
dump adios command
dump atoms/adios command
dump custom/adios command
dump cfg/uef command
dump h5md command
dump image command
mp movie command
modify command
atom vector = id, mass, type, mol, x, y, z, vx, vy, vz
compute references = c_ID, c_ID[i], c_ID[i][j], C_ID,
fix references = f_ID, f_ID[i], f_ID[i][j], F_ID, F_ID
variable references = v_name, v_name[i]
Examples
variable x index run1 run2 run3 run4 run5 run6 run7 run8
variable LoopVar loop $n
variable beta equal temp/3.0
variable b1 equal x[234]+0.5*vol
variable b1 equal "x[234] + 0.5*vol"
variable b equal xcm(mol1,x)/2.0
variable b equal c_myTemp
variable b atom x*y/vol
variable foo string myfile
variable foo internal 3.5
variable myPy python increase
variable f file values.txt
variable temp world 300.0 310.0 320.0 ${Tfinal}
variable x universe 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
variable x uloop 15 pad
variable str format x %.6g
variable x delete
NPTEL



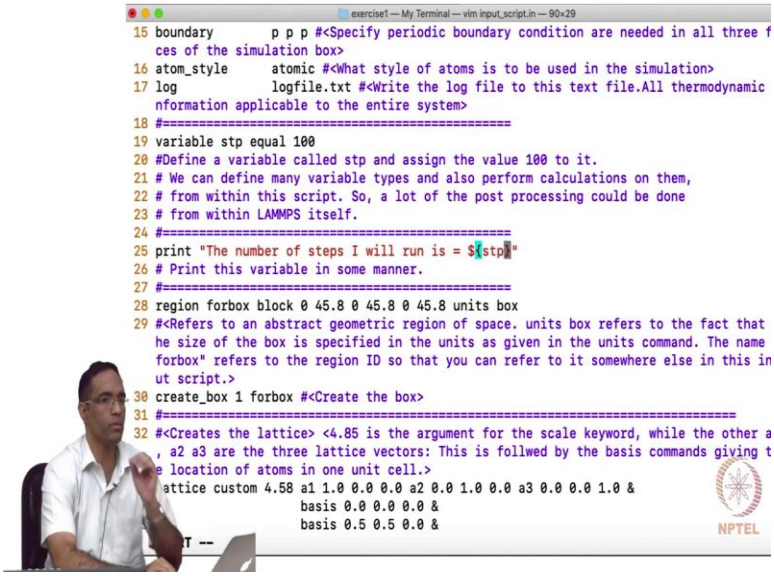
So, then we have a command call log, which basically tells the file name in which the log file of this particular simulation is going to be printed out. I am just giving some arbitrary name log file dot txt and I am going to illustrate some important thing by actually having this command right here. Then it is possible in LAMMPS for you to define variables, you can have variables either your own variables, or you can have variables that are calculated by LAMMPS. And then do some manipulations with in the input script and print them out. So, the advantage of having such an option is previously such options were not available, many years ago. So, every time you have to actually print out the coordinates, the only thing that will print out is the coordinates and probably the velocities.

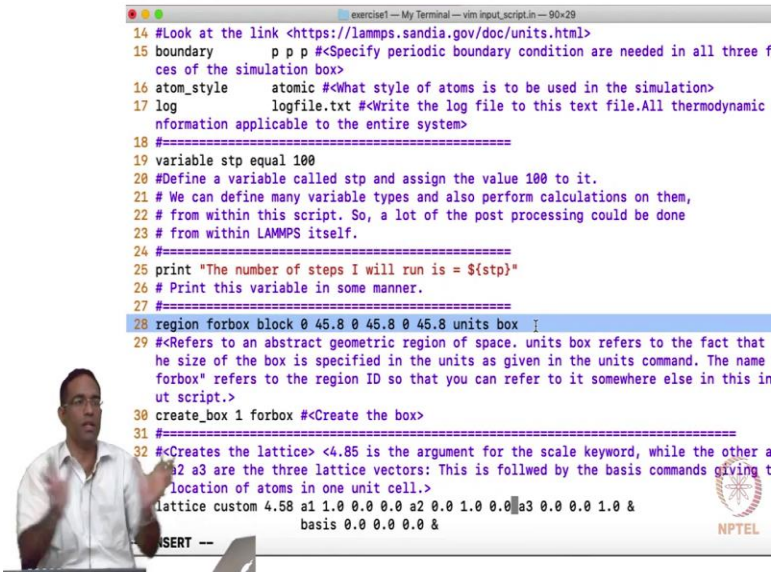
And you will have to perform a post-processing in order to determine other, other things. But now as it does evolve, the variable option is available, which allows you to actually perform manipulations on some of the variables that LAMMPS calculates, or you can do new calculations on them. So, this is actually a lot that such a lot of stuff that you can do with variables. So, we would not be able to cover all of them now. So, there is a huge number of examples, so you can have either a huge number of math functions that are available, some special functions are available.

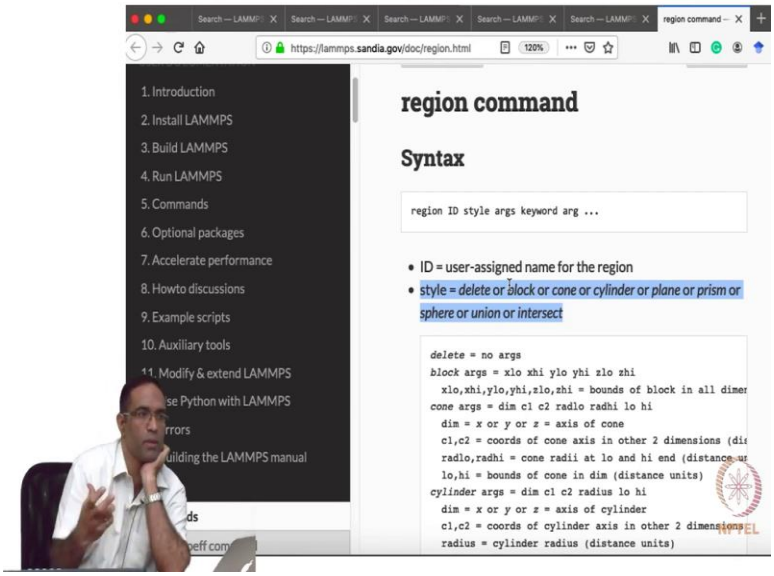
You can perform say for example you can say some calculations here, you can say that the variable b1 is equal to some x 234 probably means the x coordinate of the 234 atom plus half times the volume, and the volume is actually a keyword which LAMMPS understands to mean

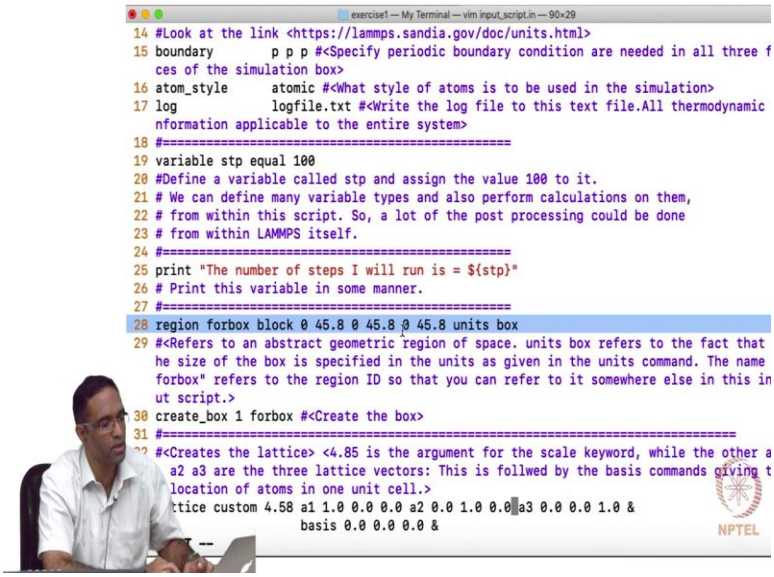
the volume of the system. And that b1 will actually store that variable. So, there can be a variable called myPy, where this variable can perform a python calculation as done by, as done by a python script. So, this variable command is pretty powerful. So, it might come of use, if you are doing some research problem in the future. But we are not going to look into such advanced stuff in this course right now. If time permits maybe for now what I am just going to demonstrate is a very simple thing.

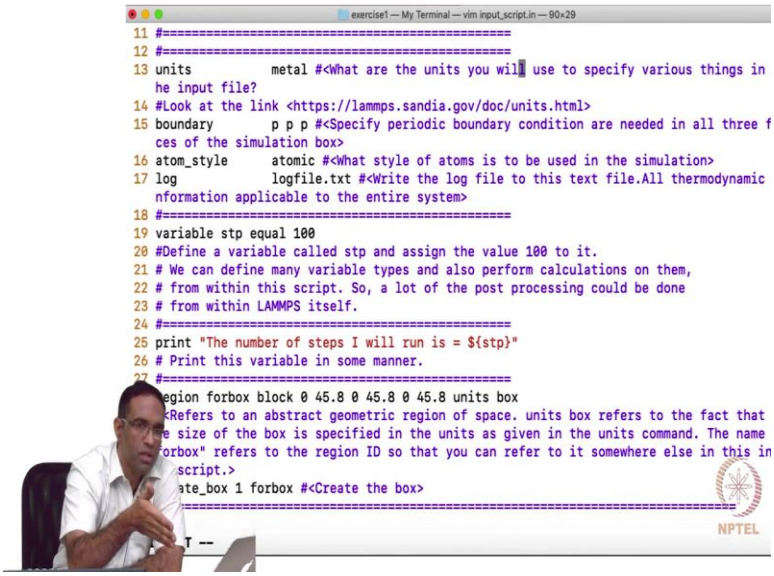
(Refer Slide Time: 18:09)

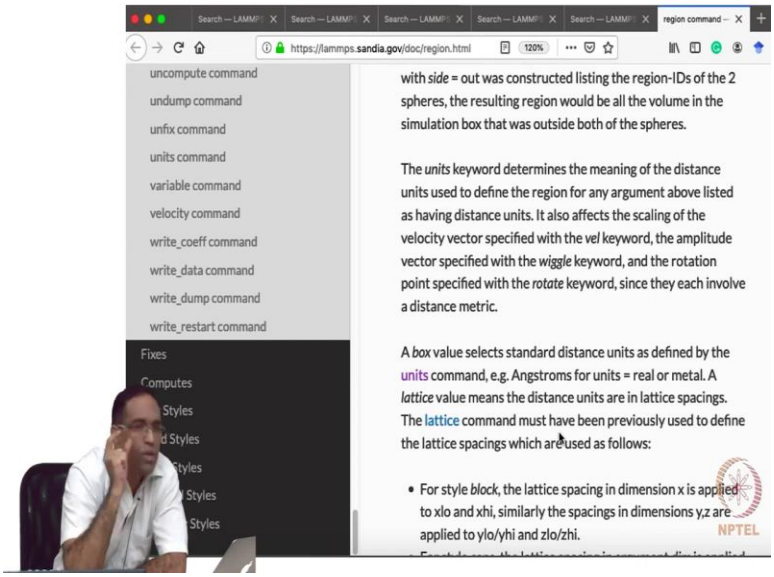





region command
https://lammps.sandia.gov/doc/region.html
1. Introduction
2. Install LAMMPS
3. Build LAMMPS
4. Run LAMMPS
5. Commands
6. Optional packages
7. Accelerate performance
8. Howto discussions
9. Example scripts
10. Auxiliary tools
Syntax
region ID style args keyword arg ...
ID = user-assigned name for the region
style = delete or block or cone or cylinder or plane or prism or sphere or union or intersect
delete = no args
block args = xlo xhi ylo yhi zlo zhi
xlo,xhi,ylo,yhi,zlo,zhi = bounds of block in all dimer
cone args = dim c1 c2 radlo radhi lo hi
dim = x or y or z = axis of cone
c1,c2 = coords of cone axis in other 2 dimensions (di
radlo,radhi = cone radii at lo and hi end (distance ur
lo,hi = bounds of cone in dim (distance units)
cylinder args = dim c1 c2 radius lo hi
dim = x or y or z = axis of cylinder
c1,c2 = coords of cylinder axis in other 2 dimensions
radius = cylinder radius (distance units)
NPTEL


exercise1 — My Terminal — vim input_script.in — 90×29
#Look at the link <https://lammps.sandia.gov/doc/units.html>
boundary p p p #<Specify periodic boundary condition are needed in all three f
ces of the simulation box>
atom_style atomic #<What style of atoms is to be used in the simulation>
log logfile.txt #<Write the log file to this text file.All thermodynamic
nformation applicable to the entire system>
#=====
variable stp equal 100
#Define a variable called stp and assign the value 100 to it.
We can define many variable types and also perform calculations on them,
from within this script. So, a lot of the post processing could be done
from within LAMMPS itself.
#=====
print "The number of steps I will run is = ${stp}"
Print this variable in some manner.
#=====
region forbox block 0 45.8 0 45.8 0 45.8 units box
#<Refers to an abstract geometric region of space. units box refers to the fact that
he size of the box is specified in the units as given in the units command. The name
forbox" refers to the region ID so that you can refer to it somewhere else in this in
ut script.>
create_box 1 forbox #<Create the box>
#=====
#<Creates the lattice> <4.85 is the argument for the scale keyword, while the other a
a2 a3 are the three lattice vectors: This is follwed by the basis commands giving t
location of atoms in one unit cell.>
ttice custom 4.58 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 &
NPTEL





So, I am going to say variable step equal 100, that means step has the value 100. And then I am going to use a command print and say the number of steps I will run is dollar step. So, putting a dollar and putting flower brackets around the variables is a way to print it, if the variable name contains more than one letter. So, for example if I just had s here, then this would be enough, however since the variable contains more than one letter. I will have to enclose them within the flower brackets in order to end emphasize that ‘stp’ itself is actually a variable, we will see in future how I am going to use that. Region.

Student: Dollar means the value of this ‘stp’?

Professor: Yes, the value of ‘stp’. Region, so, now we have done some basic setup, we have defined our atom style, boundary, units, and all that. These, these, things can occur anywhere in the program, variable, print and all that can occur anywhere in the program. Region for, forbox block, this particular command here defines a region in which you want to perform your simulations, as the name suggests. So, let us go to that command and take a look at what that does. So, in,.. now we have introduced another command, which in addition to style and args take something called as the ID, this ID is a user assigned name for this particular region. So, that you can refer to it at a later point of time in the script. So, I have named that region as forbox, you can name it anything you want.

And the style is either delete, or block, or cone, or cylinder, or plane or prism, as you can read here, or sphere. So, what this essentially means is you can create a region in this shape, our simulation box simulation domain and at this region can actually later on be filled with atoms. So, you can create a sphere, you can fill it with atoms. So, or you can choose a region in an already constructed system and delete that region using this option region delete. So, when you talk about region you can have all these different shapes for that region, and we want to have the block region, because you know we our simulation box is really a box cuboidal box.

And we are going to choose it to be a square domain, or cube or a cube cubical domain, with specific, specific parameters, X parameter, X dimensions, Y dimensions, and Z dimensions. And for the block the argument is xlo, xhi, the lower value of the x, the higher value of the x, lower value of y, higher value of y, lower value of Z is higher value of Z and this is exactly what I have specified right here. 0, 45.8, 0 45.8, 0 45.8, and I am then saying that, the units in which these distances are being specified in the input file, these in the box units. What that means is, the units are actually in the same units as defined by the units command here, which essentially means angstrom.

You can specify the units in lattice units, for example you can have your lattice constant of a say some copper as something, say some 5 angstrom, or something like that. And you can say here units equal to lattice and whatever number you are going to specify here will be in the order of lattice units, will be scaled by that parameter that you have given for the lattice command. But the simplest and the most straightforward thing is to use the Box units. So, that you are clear

about the actual units, in which the distances are being specified and that corresponds to the metal units, that we have specified over here.

Student: So, can we use different units also?

Professor: You can use any units you want, but the whole problem is you have to understand that the problem that you are addressing is in the see for example, if you use the SI units, then if you specify mass you will have to specify in kg, and if you are specifying the mass of an atom in kg, it will be what? It will be in the order of 10 to the power minus?

Student: 31.

Professor: 31 and 30, or 29, or 28, something like that. It is going to be ridiculously small and your computer will have trouble in understanding what you are saying, because it is, it is too small. So, but when you are talking about say you do not do simulations of stars here, but when you are talking about simulations of small particles, small particles of sand or small steel balls. And each of these balls is in the order of weighing in the order of grams, then you can use CGS units to actually simulate the system. But we are looking at molecular dynamics simulations from the perspective of atoms and molecules. So, we really have to stick to, not go to SI units, or CGS units, that is going to be really inconvenient.

Student: Sir, if at the beginning I specify units as real suppose, and later on I read units box instruction. So, it will set the unit has a real only, whatever I…

Professor: Whatever, whatever for real whatever is the units of distance, it will set that, it means in this, in this case, you have anything to say?

Student: It seems it is nothing to do with that this box, depending on the box, it is nothing to deal with?

Professor: No, what it means is, the units in which I am specifying the distances are box units, it is the units in which the units command has been specified, you cannot be specific arbitrary units here, what does 45.8? It has to be an angstrom.

Student: There is something units then argument will be there, from that argument only it will be, it will be…

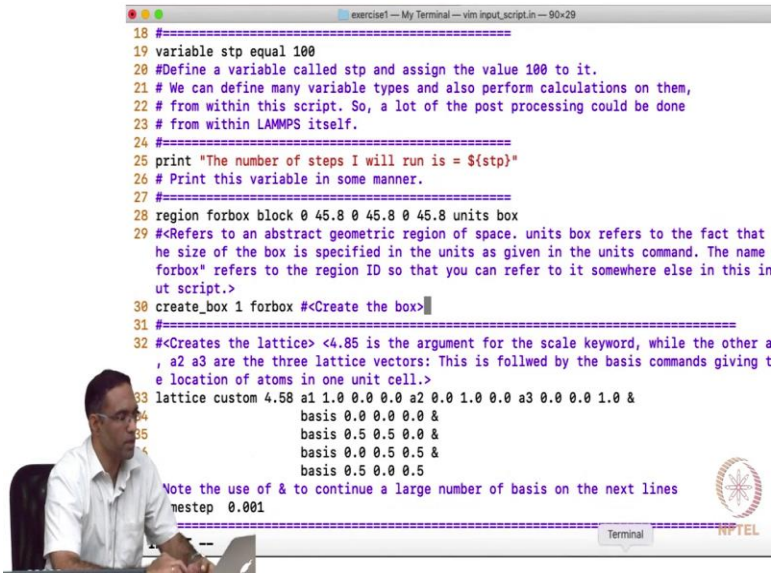
Professor: This box?

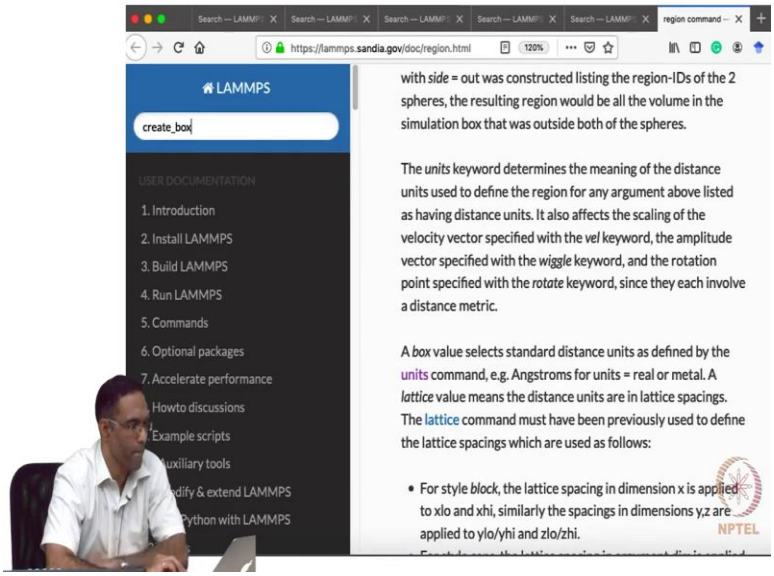
Student: Yes, it will there in the region command.

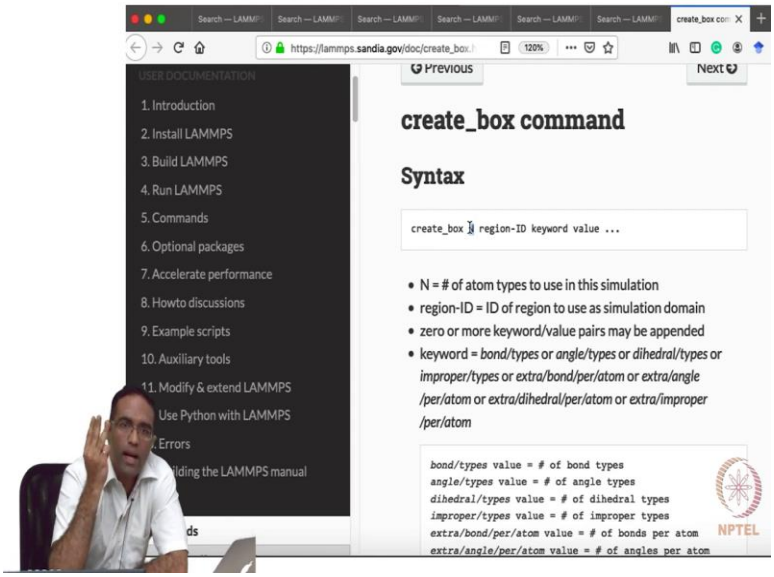
Professor: Yeah, you were already on that webpage. So, I saw this right down, so the unit's keyboard determines the meaning of the distance units used to define the region for any of the argument above listed as having distance units, it also affects the scaling of the velocity vector blah, blah, blah, blah. So, the box value selects standard distance units as defined by the units command. Angstrom units for real or a metal, a lattice value means the distance units are in terms of the lattice spacings.

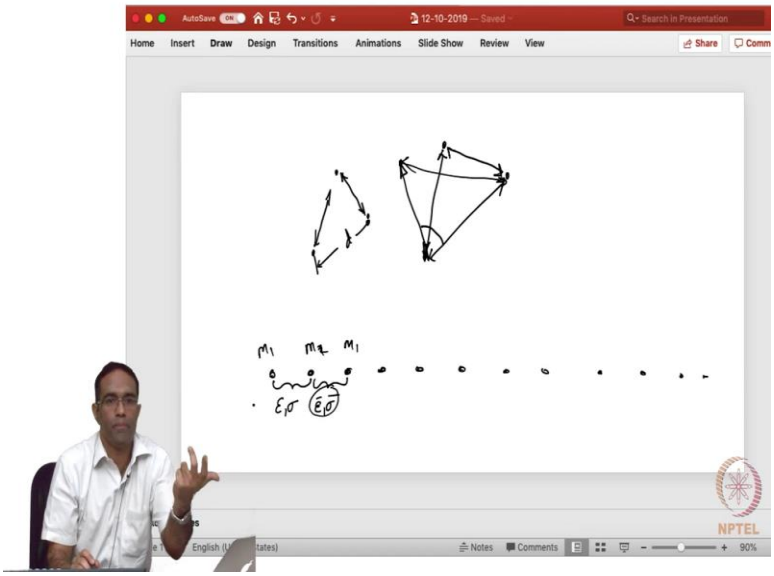
So, for example if you give their units equal to lattice 10, 10, then it essentially means if we previously somewhere if you have specified a lattice parameter of 5, that means it is going to be 50 angstroms, 50 angstroms and 50 angstroms, that is what it means. So, you have to be careful here, about what units you are using, unit is equal to box is safest, because you it is directly connected with the units command that you have specified previously, is that okay? Any questions?

(Refer Slide Time: 26:33)




https://lammps.sandia.gov/doc/region.html
LAMMPS
create_box
USER DOCUMENTATION
1. Introduction
2. Install LAMMPS
3. Build LAMMPS
4. Run LAMMPS
5. Commands
6. Optional packages
7. Accelerate performance
Howto discussions
Example scripts
uxiliary tools
dify & extend LAMMPS
ython with LAMMPS
with side = out was constructed listing the region-IDs of the 2 spheres, the resulting region would be all the volume in the simulation box that was outside both of the spheres.
The units keyword determines the meaning of the distance units used to define the region for any argument above listed as having distance units. It also affects the scaling of the velocity vector specified with the vel keyword, the amplitude vector specified with the wiggle keyword, and the rotation point specified with the rotate keyword, since they each involve a distance metric.
A box value selects standard distance units as defined by the units command, e.g. Angstroms for units = real or metal. A lattice value means the distance units are in lattice spacings. The lattice command must have been previously used to define the lattice spacings which are used as follows:
For style block, the lattice spacing in dimension x is applied to xlo and xhi, similarly the spacings in dimensions y,z are applied to ylo/yhi and zlo/zhi.
NPTEL


https://lammps.sandia.gov/doc/create_box.
Previous
Next
USER DOCUMENTATION
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
Use Python with LAMMPS
Errors
ilding the LAMMPS manual
create_box command
Syntax
create_box N region-ID keyword value ...
N = # of atom types to use in this simulation
region-ID = ID of region to use as simulation domain
zero or more keyword/value pairs may be appended
keyword = bond/types or angle/types or dihedral/types or improper/types or extra/bond/per/atom or extra/angle /per/atom or extra/dihedral/per/atom or extra/improper /per/atom
bond/types value = # of bond types
angle/types value = # of angle types
dihedral/types value = # of dihedral types
improper/types value = # of improper types
extra/bond/per/atom value = # of bonds per atom
extra/angle/per/atom value = # of angles per atom
NPTEL



Now, I just create the box, so I have defined a region of the box and now I created the box. So, create box, in a create box command we will see what are the parameters are it takes, it takes following the create box command takes N, where N is basically the total number of atom types that you are going to have, for example if you are going to have calcium fluoride, then the total number of atom types will obviously be 2, one is calcium, one is F. So, now that, that being said it can also mean different things.

So, the element may be same, so for one type of calcium you may want one specific kind of interaction with fluorine, whereas for another calcium you may want another type of interaction with the fluorine atom, just, just for the sake of example. Then you may have 3 atom, 3 atom types, but 2 of them are actually the same element truly, you understand what I am saying? No. So, it is not necessary that the atom types are actually related only to the element that is being used, the atom type can also be of the same element but you may want to specify 2 different atom types simply, because the interaction between the same element with some other element is different.

Student: Isotopes.

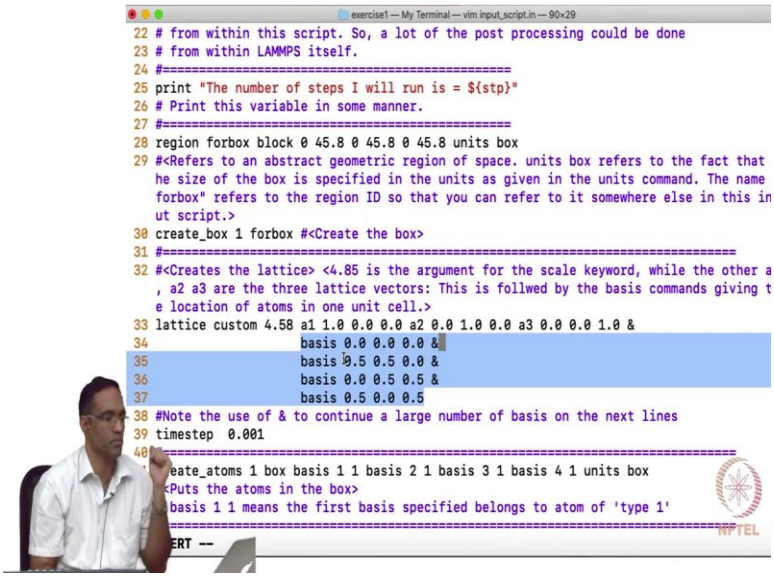
Professor: Maybe, yeah, okay. So, that is sort of a, for that sort of a thing you need to remember that the atom types are specified keeping in mind the kind of interactions and the kind of elements that are going to be used in your system.

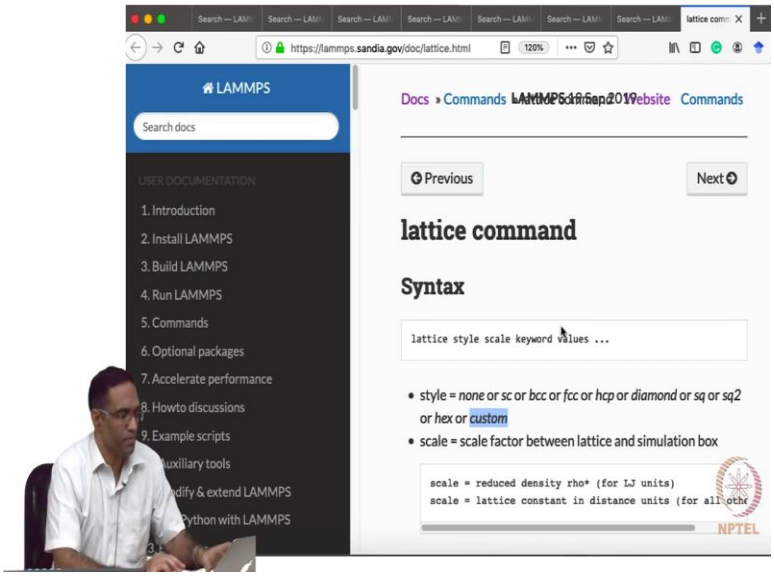
Student: Just now means can two different calcium atoms?

Professor: No, no, sorry it is just a random arbitrary example. So, suppose you have, they are all mass m1, they are all masses, are the same m1. This atom is interacting with one set of LJ parameters. Now, this atom is interacting with another set of LJ parameters, although they are calcium for some other, for some arbitrary reason I want to simulate the interaction of this calcium atom with this calcium atom through this set of LJ parameters. Then you would not be specifying all the 3 atoms are of type atom type 1, because you are going to specify the interaction between the atoms based on the types. So, you may want to specify different types even though you are actually simulating a system with only 1 atom.

So, create one forbox, so where we were looking at this thing yeah. So, it creates the region, so following, following the number of atom types that you need in the simulation, you need to specify the region ID, the region ID is something that we just specified which is called as some forbox right here, that is the region ID that is what we are using, we are saying create box, 1 total number of atom types is 1 and forbox is basically the region ID, it creates the box. So, once we create the box, we have to fill this with atoms. So, first we use a lattice command in order to specify the basis and the corresponding lattice vectors associated with our crystal structure, in the following manner.

(Refer Slide Time: 30:30)





So, lattice is another command that can be used. So, what, what this essentially means is? It is not necessary to actually use your MATLAB script to generate the lattice structures, you do not have to generate the entire lattice for your simulation purposes through an external core, you can do it from within LAMMPS itself, which allows you, which allows you to specify what are the various lattice vectors.

So, a 1, a 2 and a 3 are the various lattice vectors. So, in this case I am saying it is 1, 0, 0, 0, 1, 0, and 0, 0, 1, I am using an ampersand sign, because I am going to give a I am going to list the various basis associated with this crystal structure in subsequent lines. So, I said basis, 1 basis is at 0, 0, 0, the other one is at 0.5, 0.5, 0, other one at 0, 0. 5, 0.5, the other one is 0. 5, 0.5. So, obviously this is generating a face centered structure, 4.58 is a scaling parameter. So, basically all these values are scaled by 4.58.

So, instead of I could give here 1, I could give here 1, and replace this by 4.58. So, in this manner I have actually specified the nature of the lattice, the lattice vectors and the corresponding basis. Now, in complicated systems, in complicated systems the bases are not this simple. So, you would still need the understanding of Wyckoff position and space groups and the using and and to properly use your MATLAB script in order to generate the large number of basis which may occur for your extra structure, although the lattice vectors are the same, this set needs to be generated from that knowledge, you agree with that. So, this is exactly the same

basis, if it is going to be some complicated structure that I gave you for your assignment lithium meta-titanate, then it is going to be the huge how many atoms are there in one unit cell?

Student: 48.

Professor: 48. So, those 48 atoms will actually be listed here, you have to generate it separately, cut and paste it carefully here. Now, the lattice….

Student: It means we need to be scale the basis is also...

Professor: The basis is usually given in fractional coordinates, they are just the fractional coordinates.

Student: So, we can, I mean we can understand 4.58 can be the lattice constant

Professor: It is a lattice, it is a lattice constant, lattice constant, lattice constant.

Student: What is custom?

Professor: I am just about to go there. So, style equal to none, simple cubic, body-centered cubic, face-centered cubic, hcp, diamond, just a square, I do not know what is sq 2, I do not know what that is, hex or custom. So, if you want to generate a face centered cubic structure, you just have to say lattice FCC and the lattice constant, it will generate a face centered, because it knows what the basis are for a face centered structure. You do not even have to give the basis.

But I like to give the custom command, the reason why I use the custom command is to show you that, you can create a triclinic structure here, you see that there is no style called triclinic, or monoclinic, you know anything else. So, if you want to create a complicated structure triclinic, then you have to use the explicit lattice vectors a1, a2, and a3, in order to construct your lattice. But if you just wanted to construct a simple cubic BCC or FCC or HCP or a diamond structure, you just can give lattice space diamond space the corresponding lattice constant and it will just construct it, you do not need to give the basis. Because for these crystal types the basis is known, correct anything else, any questions so far?

Student: So the lattice vectors always have to be written as a 1, a 2, a 3?

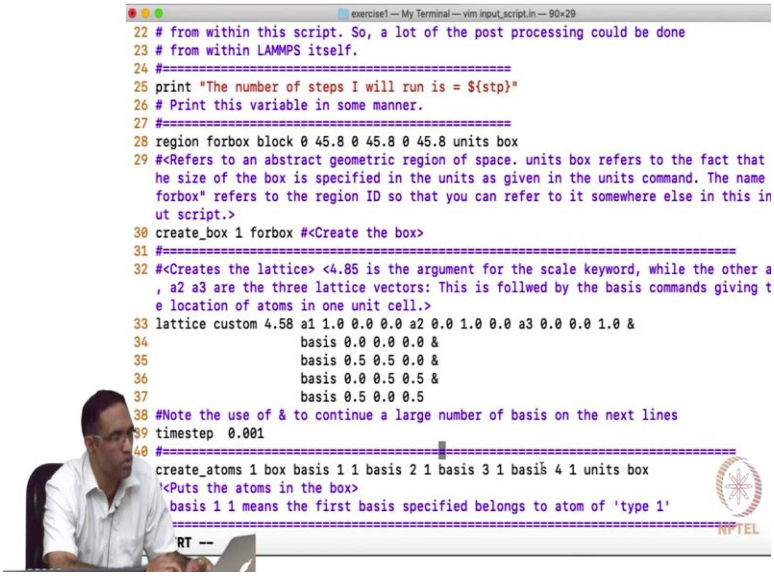
Professor: Yes.

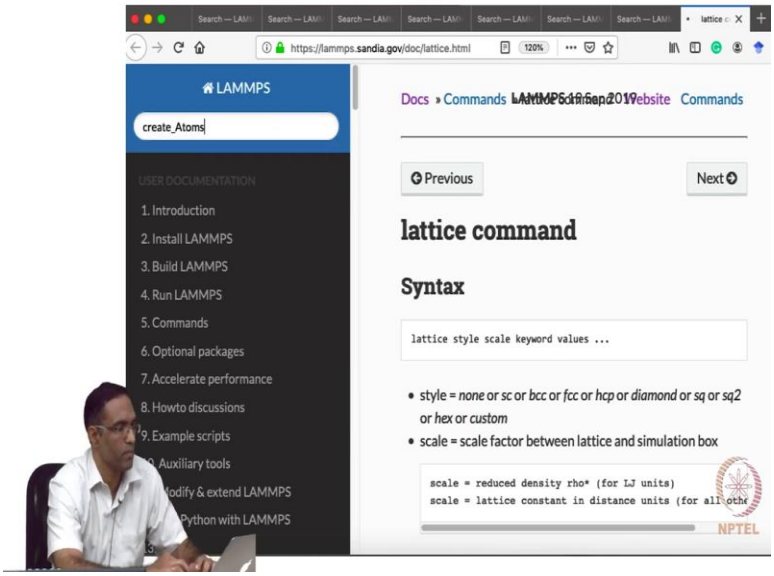
Student: Always?

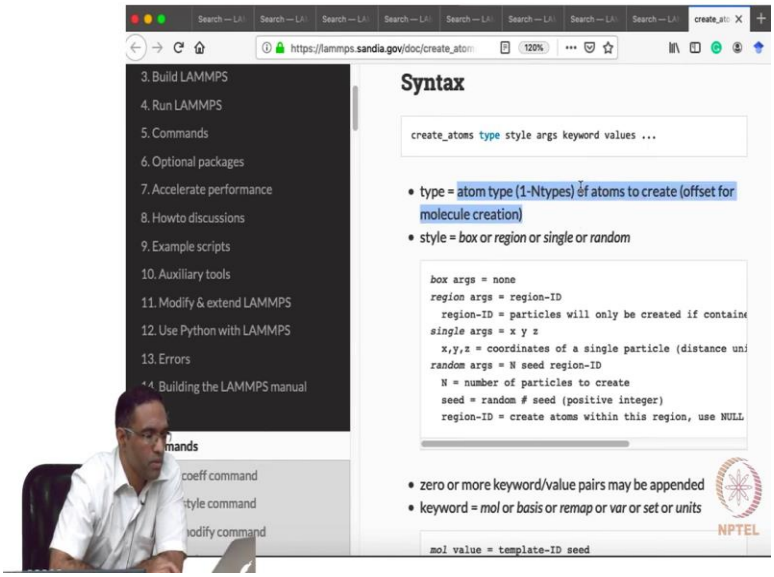
Professor: Yes, a 1, the X, X component, Y component, Z component, a 2 the corresponding, X component, Y component, Z component and so on. What alpha, beta, gamma? That is what you specify all, you specify all those alpha, beta, gamma and other things through the a 1, a 2, a 3. So, you should, you should know those values alpha, beta, gamma values to actually come up with a 1, a 2, and a 3, is not it? a 1, a 2, and a 3 are vectors in Cartesian space, they are not, they are vectors in Cartesian space.

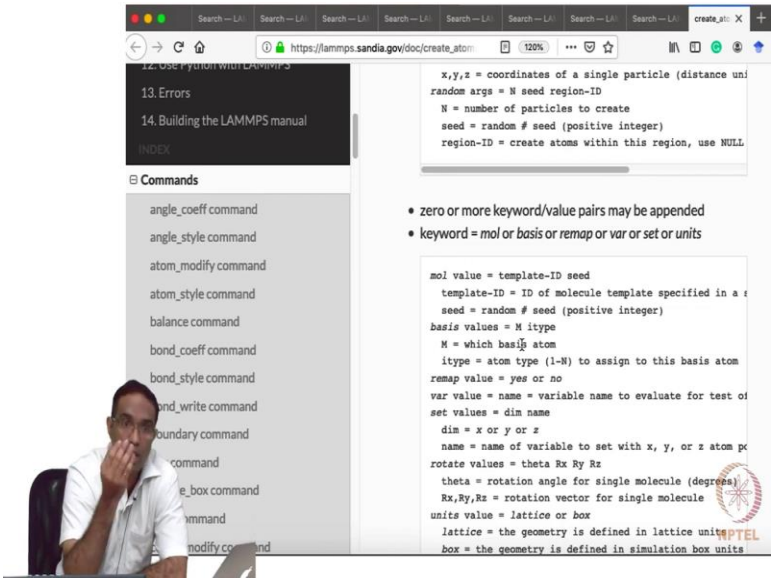
So, that the a 1, a 2, a 3, we involve that alpha, beta, gamma for example if you take a look at one of the crystal structures that we generated, monoclinic crystal structure we generated, we used that beta the angle between c and a in order to find out the components of the various components of the lattice vectors is that right, you remember, that I hope so. So, we will stop with creating this, so this one this command we will talk a little bit about time step a little bit maybe in the next class.

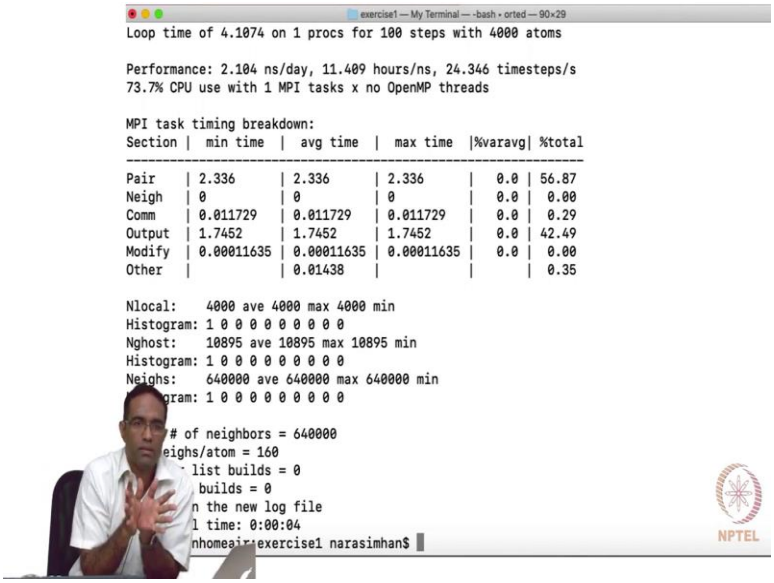
(Refer Slide Time: 36:27)

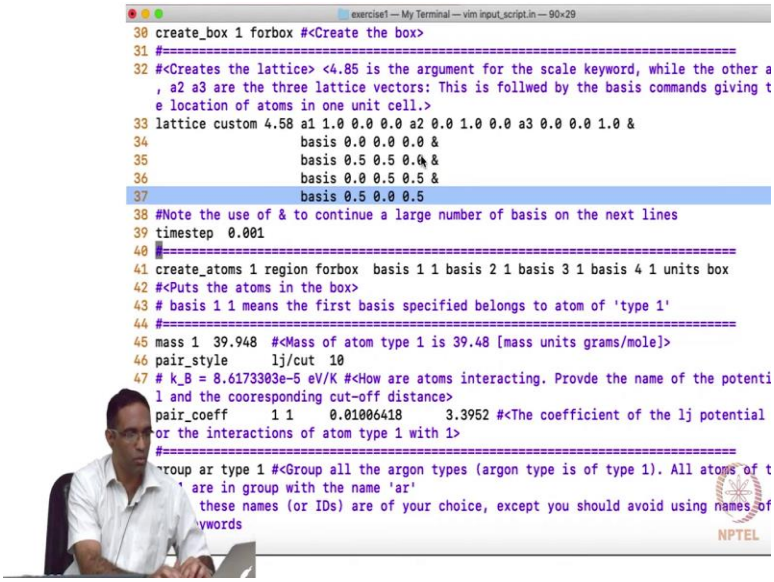


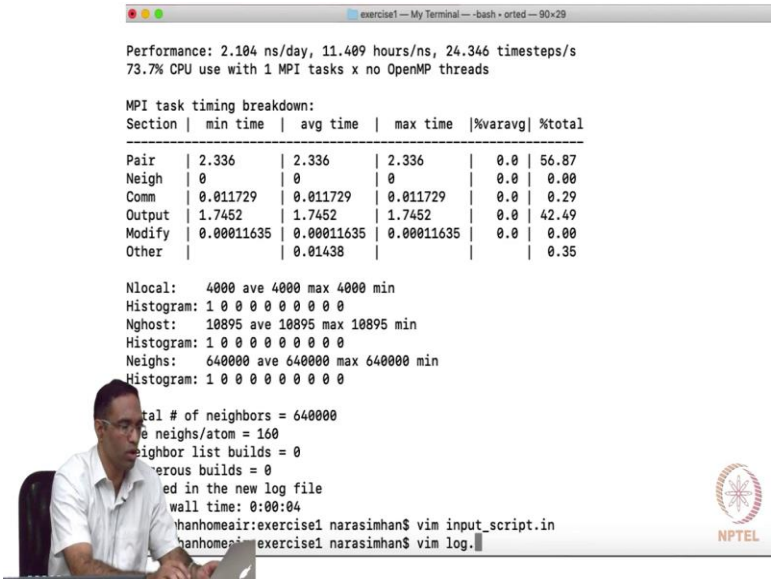

https://lammps.sandia.gov/doc/lattice.html
LAMMPS
create_Atoms
USER DOCUMENTATION
1. Introduction
2. Install LAMMPS
3. Build LAMMPS
4. Run LAMMPS
5. Commands
6. Optional packages
7. Accelerate performance
8. Howto discussions
9. Example scripts
10. Auxiliary tools
Modify & extend LAMMPS
Python with LAMMPS
Docs » Commands Website Commands
Previous
Next
lattice command
Syntax
lattice style scale keyword values ...
style = none or sc or bcc or fcc or hcp or diamond or sq or sq2 or hex or custom
scale = scale factor between lattice and simulation box
scale = reduced density rho* (for LJ units)
scale = lattice constant in distance units (for all othe
NPTEL


https://lammps.sandia.gov/doc/create_atom
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
Building the LAMMPS manual
coeff command
style command
odify command
Syntax
create_atoms type style args keyword values ...
type = atom type (1-Ntypes) of atoms to create (offset for molecule creation)
style = box or region or single or random
box args = none
region args = region-ID
region-ID = particles will only be created if containe
single args = x y z
x,y,z = coordinates of a single particle (distance uni
random args = N seed region-ID
N = number of particles to create
seed = random # seed (positive integer)
region-ID = create atoms within this region, use NULL
zero or more keyword/value pairs may be appended
keyword = mol or basis or remap or var or set or units
mol value = template-ID seed
NPTEL


https://lammps.sandia.gov/doc/create_atom
13. Errors
14. Building the LAMMPS manual
Commands
angle_coeff command
angle_style command
atom_modify command
atom_style command
balance command
bond_coeff command
bond_style command
x,y,z = coordinates of a single particle (distance uni
random args = N seed region-ID
N = number of particles to create
seed = random # seed (positive integer)
region-ID = create atoms within this region, use NULL
zero or more keyword/value pairs may be appended
keyword = mol or basis or remap or var or set or units
mol value = template-ID seed
template-ID = ID of molecule template specified in a
seed = random # seed (positive integer)
basis values = M itype
M = which basis atom
itype = atom type (1-N) to assign to this basis atom
remap value = yes or no
var value = name = variable name to evaluate for test of
set values = dim name
dim = x or y or z
name = name of variable to set with x, y, or z atom pc
rotate values = theta Rx Ry Rz
theta = rotation angle for single molecule (degrees)
Rx,Ry,Rz = rotation vector for single molecule
units value = lattice or box
lattice = the geometry is defined in lattice units
box = the geometry is defined in simulation box units
NPTEL


exercise1 — My Terminal — -bash • orted — 90×29
Loop time of 4.1074 on 1 procs for 100 steps with 4000 atoms
Performance: 2.104 ns/day, 11.409 hours/ns, 24.346 timesteps/s
73.7% CPU use with 1 MPI tasks x no OpenMP threads
MPI task timing breakdown:
Section | min time | avg time | max time |%varavg| %total
Pair | 2.336 | 2.336 | 2.336 | 0.0 | 56.87
Neigh | 0 | 0 | 0 | 0.0 | 0.00
Comm | 0.011729 | 0.011729 | 0.011729 | 0.0 | 0.29
Output | 1.7452 | 1.7452 | 1.7452 | 0.0 | 42.49
Modify | 0.00011635 | 0.00011635 | 0.00011635 | 0.0 | 0.00
Other | | 0.01438 | | | 0.35
Nlocal: 4000 ave 4000 max 4000 min
Histogram: 1 0 0 0 0 0 0 0 0 0
Nghost: 10895 ave 10895 max 10895 min
Histogram: 1 0 0 0 0 0 0 0 0 0
Neighs: 640000 ave 640000 max 640000 min
Histogram: 1 0 0 0 0 0 0 0 0 0
# of neighbors = 640000
eighs/atom = 160
list builds = 0
builds = 0
n the new log file
l time: 0:00:04
nhomeair:exercise1 narasimhan$
NPTEL





Create_atoms command basically specifies where you want to create these atoms. So, it says create_atoms 1 box, and then basis 1 is of type 1, basis 2 is of type 1, basis 3 is also of type 1, and basis 4 is also of type 1. So, let us see what this create atoms contains. Atom type, atom type information, 1 to the total number of types of atoms that you want to generate, it is a box or is it a region, or is it a single, or a random. So, we want to actually use region ID, so I should have used region ID here, I just used box. So, I just used box here, but it is possible for us to use this region ID also in order to create this atoms, is it right Deepak? So...

Student: Region which are using two regions do not specify.

Professor: You not specify which region you want to put the atoms in. So, in this case since there is only 1 region, it is not essential that we created a region, we could have just given the type as a as box and created and given the list of the basis the corresponding right here. So, basis value is equal to M i type. So, base m equal to which basis atom? So, you specified several basis, when you define the lattice vector. So, the first one that you specify the basis atom 1, the second one is basis atom 2, and third one is basis atom 3, and so on. So, you have to specify what is a type of each basis atom. So, there are 1 there is only 1 type in this particular simulation.

So, but if you are basis atom involves 48 different atoms, if there are 48 different basis within your unit cell, there is a lithium, titanium, and oxygen, then lithium may be basis atom 1, titanium may be basis atom 2, oxygen may be basis atom 3 and so on. You need to know that list and that list has to be specified here, this is what basis 1, basis 2, 1, basis 3,1 basis 4, 1. So, this probably may be confusing, so what I am just going to do, I am going to try to change this to region. So, I am going to say instead of box I am going to say region followed by the region ID. So, let us see whether that works, what is the name of the region, region ID is yeah. So, so this the region in which the atoms have to be created is what is given there. So, are there any questions so far?

Student: Sir if I had a iron …. So it would simulating with BCC structures having only iron atoms, that is it. So could you hold on, if that line we have to write, basis 1, 1 basis 2, 1.

Professor: Which line?

Student: Sir that a atoms.

Professor: Basis 2, 1 yes, yes,

Student: Basis 1 1, basis 2 1, it is first atom is on first step, second atom is also...

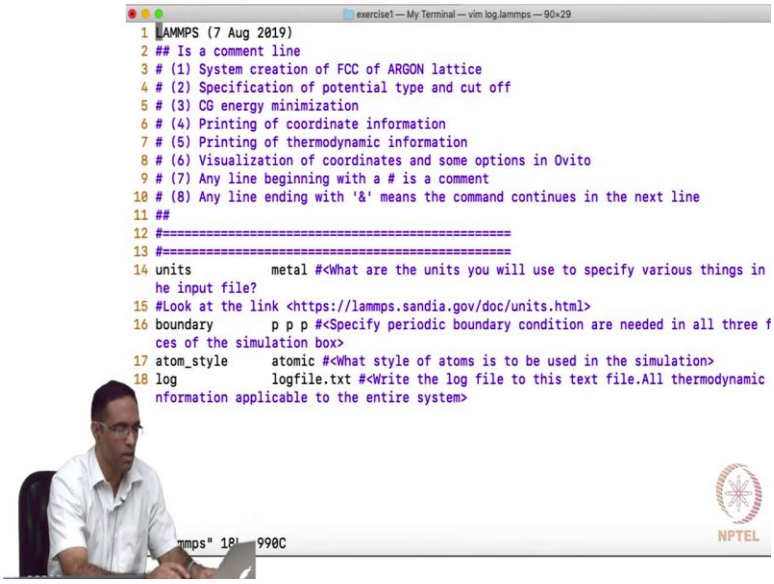
Professor: First, first step. That’s it.

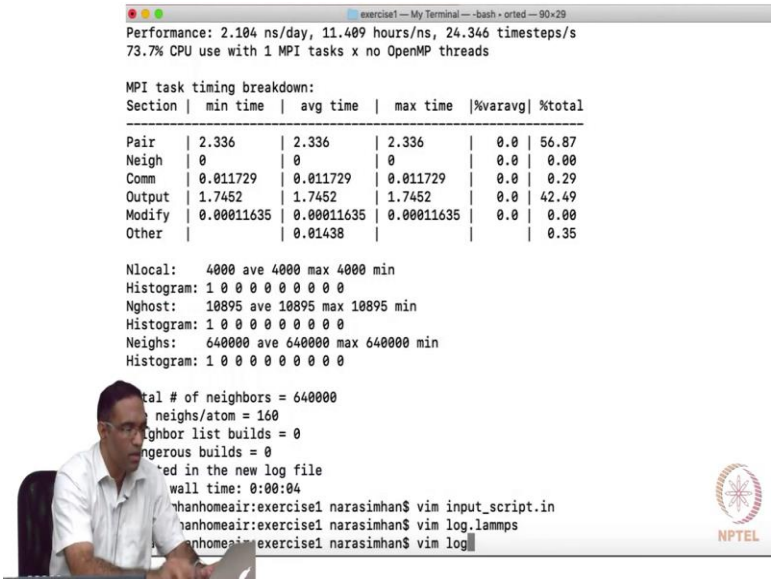
Student: Okay only iron...

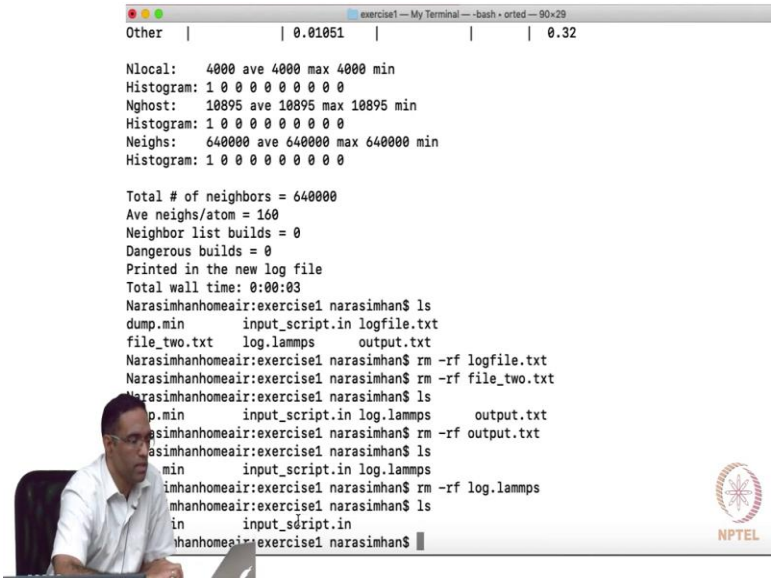
Professor: Only iron is there, but if you have other kinds of, iron carbon if you are simulating the first atom may be of type iron and the second atom may be of type carbon, and that basis corresponds to this list, this list, this is, this is basis, this is basis 1, this is basis 2, this is basis 3,

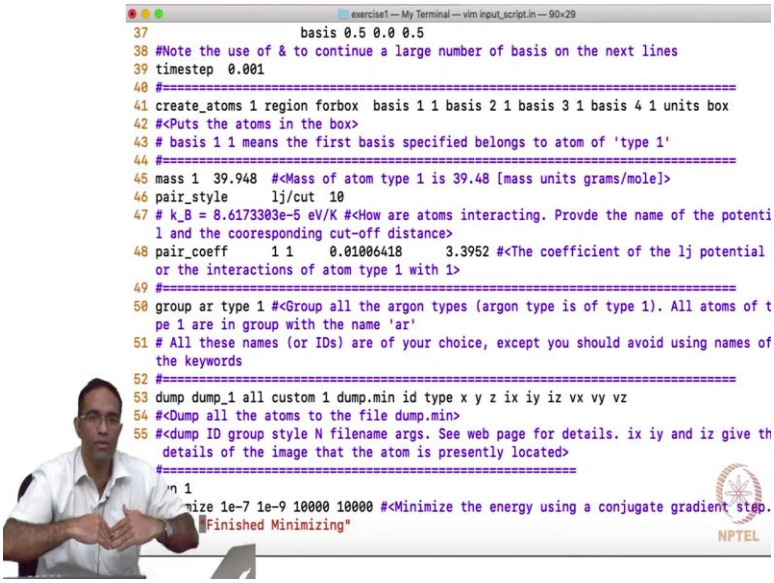
this is basis 4 and so on. And that is this 1 maps to this 1, this 2 maps to this 2, this 3 maps to this 3, and this 4 maps to this 4.

(Refer Slide Time: 41:28)









Now, the most important thing in LAMMPS is: LAMMPS execute the we will look at the other codes in the next class, other part of the code of the next class, but I want to end the class by saying that LAMMPS execute things are line by line, it reads a line executes it, it reads a line executes it and so on. So, the problem is say you made a mistake, so when I ran this, when I ran this code it prints out some log file by default it is called as log dot LAMMPS. And it prints something out, suppose I make a mistake.

So, you see that the, the log information stops at this file log, log file dot txt, it stops here the log information that it is printing out stops at the line log log file dot txt and continues in the file log file dot txt, you following what I am saying, no? Okay good. So, the file that I need to run an, run

a LAMMPS program are the input script, that is it and this dump min is actually an output that it has printed out.

So, let me run it again, so it is printing out some information to the screen, that information is actually going to be there in log dot LAMMPS, the entire information is there in log dot LAMMPS. It has a lot of other a lot of information we will go through this maybe in the next class. Now, what I am going to do is in the input script, I am going to say stop dumping in log dot LAMMPS, but start dumping in log file dot txt. So, if I run LAMMPS again, what it is going to do is, in log dot LAMMPS previously you saw the entire output. But now you are only seeing it until the line log log file dot txt, the reminder is actually printed in log file dot txt.

So, this is I mean nobody would do such things, I just wanted to demonstrate that it reads the line and executes it. So, the whole problem with that is if you did not realize it what would happen is? If you made an error somewhere here somewhere down the script, then you are not going to find it out until LAMMPS has finished reading all the lines until before the point. So, say you had two different runs in your input script, 1 is running for 1, 1 set of runs takes a day, the second set takes about half a day. And if you made a mistake in this second part, you know that you made a mistake in the second part only after it run for 1 full day.

So, that is why you need to be extremely careful in writing this input script for LAMMPS, it is not like a compiled program that it will like a C++ compiler it will take a look at all your entire input file and then tell you, you forgot a semicolon here or something. It is not going to do that, it is going to wait for, it is going to run everything above it before it finds that error. So, when you are running large extremely large simulations the idea is to run every single part of your code for just one step quickly to see if it is going through the entire input script and then increase your runs to a larger number of times steps.