Two Phase Flow and Heat Transfer

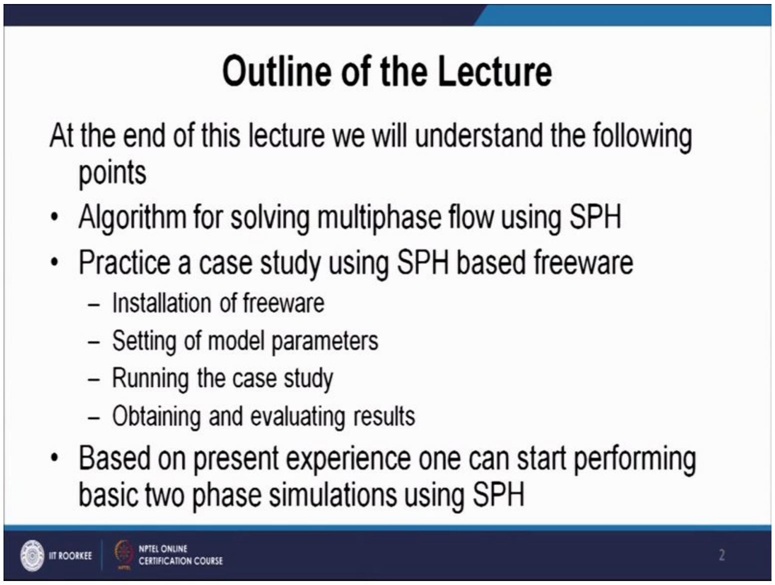
Dr. Arup Kumar Das
Department Of Mechanical and Industrial Engineering,
Indian Institute of Technology, Roorkee

Lecture No: 15
Smoothed Particle Hydrodynamics

Hello welcome in the fifteenth lecture of Two Phase Flow and Heat transfer. Today we will be discussing about Smoothed Particle Hydrodynamics. This is Lagrangian methodology. So let us see at the end of this lecture. We will be understanding the algorithm for solving multiphase flow in SPH; we will be doing a practice of this type of particle based methodology simulation using a freeware. So we will be installing the freeware.

We will be showing you how to set the model parameter, how to run the case study and finally how to obtain the result and analyze those.

(Refer Slide Time: 00:45)



Outline of the Lecture

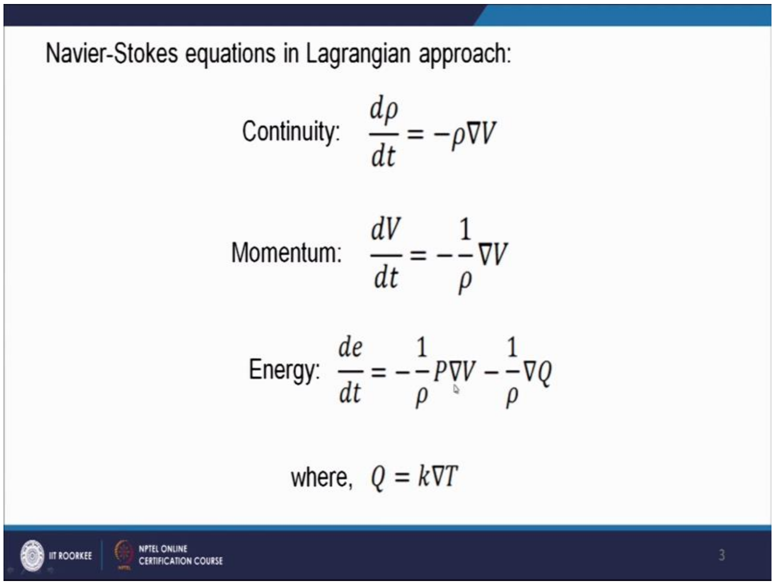
At the end of this lecture we will understand the following points

- Algorithm for solving multiphase flow using SPH
- Practice a case study using SPH based freeware
  - Installation of freeware
  - Setting of model parameters
  - Running the case study
  - Obtaining and evaluating results
- Based on present experience one can start performing basic two phase simulations using SPH

Based on the present experience you can start your own simulation for Two Phase Flow using Smoothed Particle Hydrodynamics. So let us first go to the equations, what we follow in smoothed particle hydrodynamics. As I have mention Smoothed Particle Hydrodynamics is one Lagrangian type of method. So obviously we have to deal with Lagrangian equations okay.

So in Lagrangian framework, if we try to write down the Navier Stokes equation and energy equations, the equations will look like this. So continuity equations will be transforming into dRho/ dt = - Rho del v okay. Momentum equations will be transforming to dv/ dt = - 1 /Rho del v okay.

(Refer Slide Time: 01:56)



Navier-Stokes equations in Lagrangian approach:

Continuity: $$\frac{d\rho}{dt} = -\rho\nabla V$$

Momentum: $$\frac{dV}{dt} = -\frac{1}{\rho}\nabla V$$

Energy: $$\frac{de}{dt} = -\frac{1}{\rho}P\nabla V - \frac{1}{\rho}\nabla Q$$

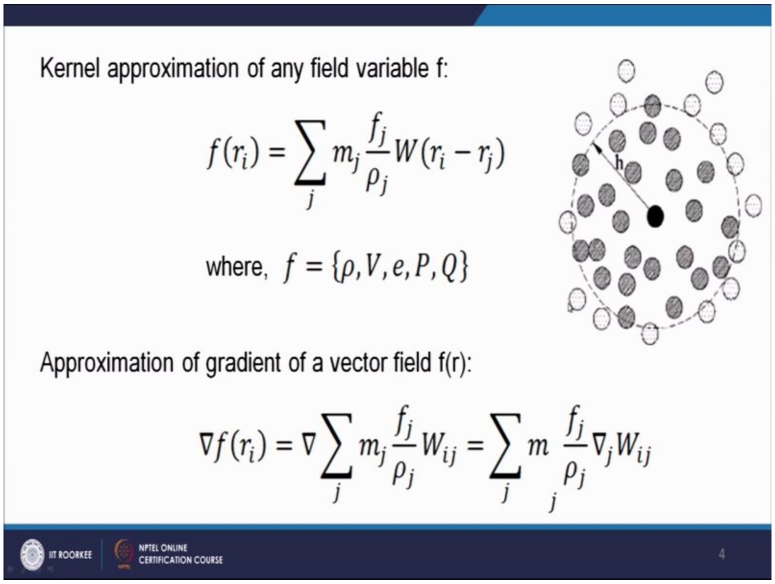
where, $Q = k\nabla T$

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
3

And then energy equation will be transforming to de/dt = - 1 / Rho p del v - 1 /Rho * del q, q is the heat transfer over here due to conduction. So q = k * delta t okay. So these equations are coming from fluid mechanics. So I am not going to elaborate discussion of these equations. So lets us see that how using smoothed particle hydrodynamics. We can tackle this type of equations especially for Two Phase Flow okay.

So in Smoothed Particle Hydrodynamics, the domain is actually discretized using Lagrangian particles. So this is the new concept in our common computational fluid dynamics analysis we have seen the domain is discretized into either volume or element or nodes in finite difference okay.

(Refer Slide Time 03:04)


h
IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
4

Kernel approximation of any field variable f:

$$f(r_i) = \sum_j m_j \frac{f_j}{\rho_j} W(r_i - r_j)$$

where, $f = \{\rho, V, e, P, Q\}$

Approximation of gradient of a vector field f(r):

$$\nabla f(r_i) = \nabla \sum_j m_j \frac{f_j}{\rho_j} W_{ij} = \sum_j \underset{j}{m} \frac{f_j}{\rho_j} \nabla_j W_{ij}$$

So here in case of this smoothed particle hydrodynamics, the domain is actually consisting a arbitrarily distributed particles. So here I have shown you one typical example. Let say this is your domain and in this you are having arbitrarily distributed particles okay. The particles are always doing some sort of interactions amongst themselves, what type of interactions depending on the governing equation we need to decide.

Apart from that we need to also find out that which particles are actually influencing a single particle of interest. Let say here this black particle is our particle of interest. So we can draw always a circle in 2 dimensional or a sphere in 3 dimension having radius that radius let say is h. This radius is called smoothing radius or smoothing zone okay. And then you see in this domain, circular domain the particles which are falling inside those actually will be you know influencing the particle of interest here the black particle okay.

So this h is very, very important in case of your Smoothed Particle Hydrodynamics simulation. If you give a very large h then all the particles will be coming into picture and if you give smaller one then you have to you will be finding out that your results are not coming into the converge simulation okay. So choice of h is very, very important okay. Next, why this Lagrangian field is chosen?

Let us see actually we will be trying to convert the differential equations into algebraic equations, which is the basic essence of computational fluid dynamics. So what will be doing over here? Let say you have a field which is f okay. So define at a particular position i. So, let say this is your ith position. So the position can be known as ri let say the location of this one is known as ri. So f ri is the field at ith position.

So that can be written as summation of mass of all other neighboring particles coming under the influence of these particles. So all these shaded particles are having denomination as jth particle. So mass of jth particle multiplied by field of that of those neighboring jth particles divided by Rho j, density of those jth particles multiplied by some smoothing function w. The smoothing function is actually a parameter is depends on the parameter which is nothing but the mutual distance between the particle of interest and the neighboring particle, so ri - rj right.

So the field variable we have found out depending on the values of the field variable of the neighboring particles okay. So neighboring particle j will be very, very important and based on all the j's or neighbors we have to sum it up okay. Now this field variable can be anything. It can start from Rho density, it can be velocity, it can be energy, it can be pressure, it can be heat flux anything okay.

So in our equations, previous equations we have shown that we are having density, velocity and temperature. It can be some more quantity also okay. Next, why this kind of approximation is necessary? By the way this approximation is called a Kernel approximation. And here whatever smoothing function we used that is actually called a Kernel okay.

Now why this type of Kernel approximation is important you see, if we are interested to find out the f ri, inside this you see apart from this ri - rj that means this smoothing function, we do not have anywhere the dependence of i okay. So if I try to discretize or if I try to differentiate this variable with the value of ri then I will be finding out this term will be acting mj fj/ Rho j this will be acting as constant okay. Because ri is no where present over there.

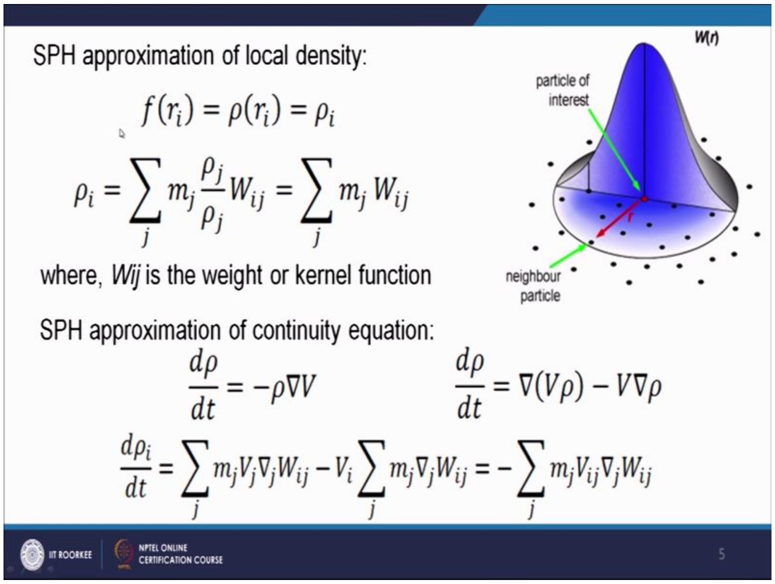
So only ri is present in the smoothing function so only the differentiation will be coming on the smoothing function. So here I have shown you that one. Let us see the first derivative you want to make with respect to a let us say ri okay for over this f ri. So you will be finding out that this term okay whatever we are having that will be acting mj * fj /Rho j that will be acting as constant.

So that can come out and only the differentiation will be acting on the smoothing function okay. w r ij , w ri - w rj okay. So that differentiations only comes over this w ij okay. So this is very important advantage why because the field variables you need not differentiate, only the differentiation of smoothing function, which we are predefining or we are presetting that one will be actually working for us okay.

So here the simulation actually boils down into book keeping off all the particles informations okay. And the differentiation on the smoothing function will do your task. Discretization in a proper fashion is not necessary. Okay, so you can go for higher order derivatives also so nth order derivative we will be giving you over here in nth order derivative of the smoothing function waste things will be coming out as constant in the equations okay.

So let us see that how this concept can be used for solving momentum equation and continuity equation. So what first will be defining, we will be first defining the local density of particle okay. So local density, let us say this f whatever I have talked about the field variable is density okay.

(Refer Slide Time: 8:32)



SPH approximation of local density:

$$f(r_i) = \rho(r_i) = \rho_i$$

$$\rho_i = \sum_j m_j \frac{\rho_j}{\rho_j} W_{ij} = \sum_j m_j W_{ij}$$

where, *Wij* is the weight or kernel function

SPH approximation of continuity equation:

$$\frac{d\rho}{dt} = -\rho\nabla V \qquad \frac{d\rho}{dt} = \nabla(V\rho) - V\nabla\rho$$

$$\frac{d\rho_i}{dt} = \sum_j m_j V_j \nabla_j W_{ij} - V_i \sum_j m_j \nabla_j W_{ij} = -\sum_j m_j V_{ij} \nabla_j W_{ij}$$

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE 5

So f ri is nothing but you Rho ri or Rho i. So f we are first considering as density. So if you try to write down what is fi or Rho i okay using the previous definition as I have shown you over here. Then we will be finding out this Rho i will be = mj * Rho j / Rho j * w of ri - rj. So Rho j Rho j can be cancelled out.

So finally we will be getting that Rho i is nothing but summation of mj * wj. Rho j Rho j is cancelling out over here okay. Now this w ij is nothing but the Kernel function okay. So density of a particle can be defined by the summation of mj *w ij okay. Now if we try to write down the continuity equation, you know that the continuity equation already I have shown you d Rho dt = - of Rho*delta v, which can be once again you know simplified as delta of v Rho - of v * delta Rho by using differentiation by part formulations.

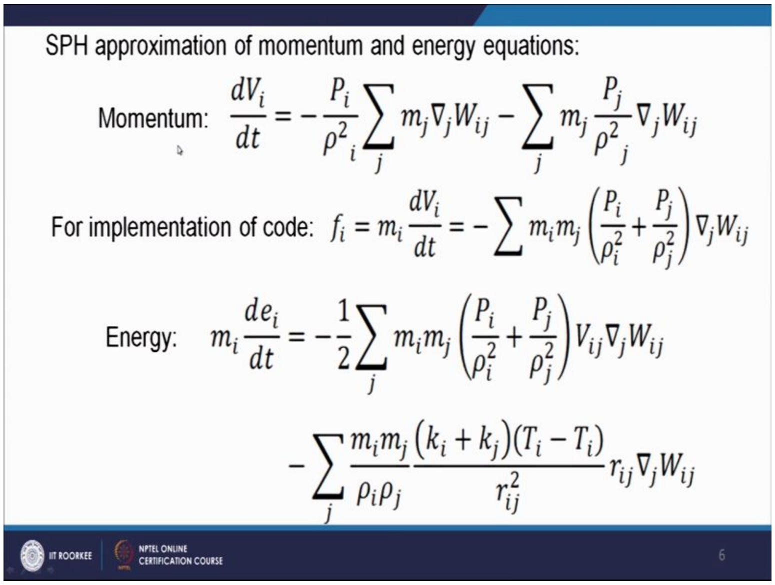
And here you see the right hand side, we will be writing here we are having the derivative of some field variable. So field derivable is v *Rho. So you can write down mj *vj Rho j / Rho j. So Rho j Rho j cancels out and then differentiation of the w because we had first derivative of v * Rho. So this will turn out first derivative of the w ij okay. Next term, over here using we are having only del Rho. So vi will be remaining outside because this we are writing for i th particle.

So for del Rho, we will be writing down. Already I have shown what is Rho i. So what del Rho i, it will be nothing but mj * del of w ij because this mj is not dependent on the i th particle okay.

So our continuity equation comes out to be d Rho i /dt is equals to here you see, here we are having vi and here we are having vj. So I can write down this is nothing but and we are having - sign over here. So you can write down this is nothing but v ij.

So finally I get d Rho i/ dt = - of summation of mj vij and then del of w ij okay. Similarly, momentum equation can be written. So I am not describing the full procedure over here but analogy will be same if you are having differentiation that differentiation can go to the smoothing function and the variable can remain outside.

(Refer Slide Time: 11:21)



SPH approximation of momentum and energy equations:

Momentum: $$\frac{dV_i}{dt} = -\frac{P_i}{\rho^2{}_i}\sum_j m_j \nabla_j W_{ij} - \sum_j m_j \frac{P_j}{\rho^2{}_j} \nabla_j W_{ij}$$

For implementation of code: $$f_i = m_i \frac{dV_i}{dt} = -\sum m_i m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2}\right) \nabla_j W_{ij}$$

Energy: $$m_i \frac{de_i}{dt} = -\frac{1}{2}\sum_j m_i m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2}\right) V_{ij} \nabla_j W_{ij}$$

$$-\sum_j \frac{m_i m_j}{\rho_i \rho_j} \frac{(k_i + k_j)(T_i - T_i)}{r_{ij}^2} r_{ij} \nabla_j W_{ij}$$

So here in a similar fashion you see this is the momentum equation. So left hand side we have kept as usual because with this we will be finding out the next time instant velocity. So dvi/dt. In the right hand side use this is coming from the pressure term. So we had del p. So del p can be written as pi Rho i square summation of mj del of wij. So, you see derivative once again coming to the smoothing function.

And here also what we can do as we are having dependence on both the particles that means your pi and pj. So you can write down this is on pj and here we have taken this 1 inside. So this is nothing but in your momentum equation if you see here we are having del v, so del v can be written as pi is going outside. So this is nothing but your del v over here and this is del vp okay.

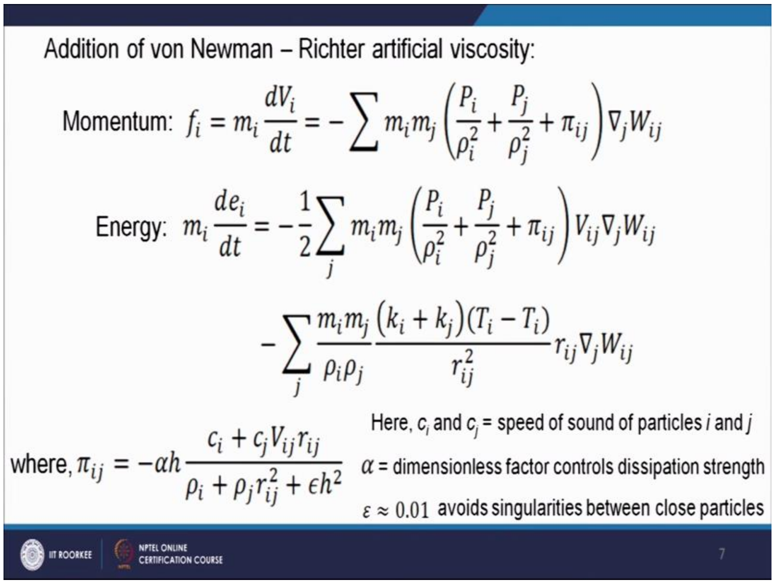
So p is taken inside using by parts okay. Now for finding out your next positions values, so we have to find out the force.

So what you will be doing this fi is nothing but mi *d vi /dt, d vi/dt already I have found out over here. So fi will become summation of mi mj and then within bracket (pi/ Rho i square + pj / Rho j square) and sole derivative will be remaining on w okay. Similarly energy equation can be written.

So this is your energy equation part, the left hand side we are having that the temporal variation of the energy and in the right hand side we are having – ½ summation of mi mj (pi/ Rho i square +pj / Rho j square) vj vij and then derivative over the w okay. This one is for the conduction term okay. So we are having ki + kj /rij * ti - tj rij into derivative of wij okay.

So this is momentum equation final momentum equation, this is final energy equation comes out to be. One important thing here we need to look is that no where the derivative of field variable is there. So only the derivative is over the smoothing function okay. Now we know that what we have done in this methodology that the domain we have discretized in particles.

So now those particles will be breaking the continuum nature of the fluid. So to gain back the continuum nature of the fluid what we will need to do, we need to give some artificial viscosity from outside okay. So here we are giving the artificial viscosity using von Newman-Richter artificial velocity.
(Refer Slide Time: 14:06)


IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
7

Addition of von Newman – Richter artificial viscosity:

Momentum: $$f_i = m_i \frac{dV_i}{dt} = -\sum m_i m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \pi_{ij} \right) \nabla_j W_{ij}$$

Energy: $$m_i \frac{de_i}{dt} = -\frac{1}{2} \sum_j m_i m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \pi_{ij} \right) V_{ij} \nabla_j W_{ij}$$

$$-\sum_j \frac{m_i m_j}{\rho_i \rho_j} \frac{(k_i + k_j)(T_i - T_i)}{r_{ij}^2} r_{ij} \nabla_j W_{ij}$$

where, $\pi_{ij} = -\alpha h \frac{c_i + c_j V_{ij} r_{ij}}{\rho_i + \rho_j r_{ij}^2 + \epsilon h^2}$

Here, c_i and c_j = speed of sound of particles i and j

α = dimensionless factor controls dissipation strength

$\varepsilon \approx 0.01$ avoids singularities between close particles

So you see over here if you remember the momentum equation earlier we had fi is = - of summation of mi mj and then (pi /Rho i square + pj / Rho j square). Here we are adding this extra term pi ij based on the von Newman-Richter artificial viscosity okay and this term was earlier also there derivative of the smoothing function right. So this is the proposition of von Newman-Richter they said that this will be the artificial viscosity we can add to gain back the continue on nature okay.

Similarly the energy equation, you see energy equation also we had this pressure terms okay. So there we can add this pi ij okay. Earlier this was with pi ij = 0 without this von Newman-Richter artificial viscosity okay. Here we can add also pij over here right. Now what is this pi ij, this von Newman and Richter they proposed pi ij can be written as - of alpha *h okay, remember this h is nothing but your smoothing length multiplied by ci + cj *vij rij / Rho i + Rho j rij square + epsilon * h square okay.

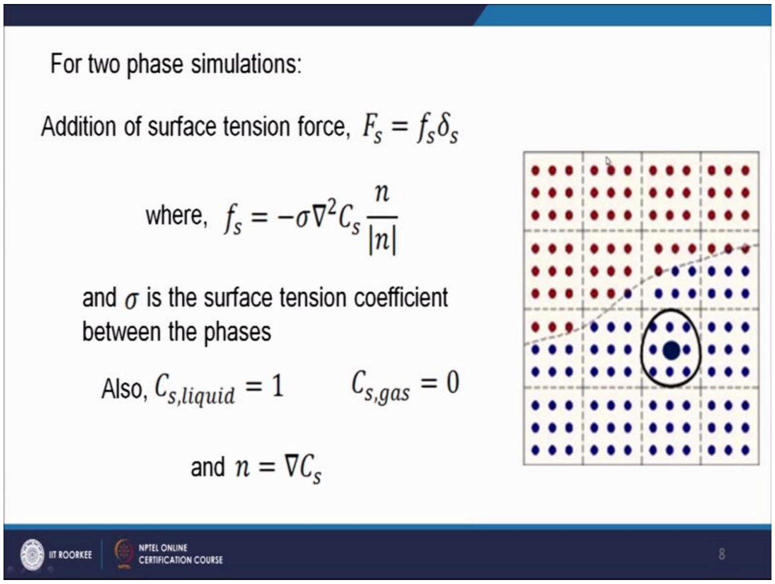
Now what are this factors ci and cj we can understand those are the sound speeds of i th and j th particle, alpha is dimensionless factor and it controls the dissipation strength. So it depends on the fluids which parameter we are taking okay. So depending on fluids the alpha needs to be tuned okay. So this is the tuning factor which we are having to control the viscosity okay. So obviously the particle let nature will not be gaining back the continuum of nature for all the fluids with the same value of alpha okay.

And epsilon which is nothing but 0.01 it avoids singularity between the close particles, because if the particles are coming very close to each other, they will be once again acting as you know continuum. So this sigma is actually given for this one okay. So once we know that what are these tuning factors okay, alpha and sigma alpha and epsilon and cij at the particle sound speeds okay, all these are known then you can find what is pi ij you can put it over here for the i and jth particle and you can evaluate the values of your fi and subsequently what is your energy in the next section okay next time step.

So in this way we will be continuing but the problem will be becoming little difficult whenever you are having 2 phase because in 2 phase, we know there is some interface and across the interface will be having some surface tension force applicable. So what we do to take care of that one in this SPH simulation, we actually denote the particles with some color okay. So these colors are dependent on the phases.

So let us say we are having 2 phases, gaseous phase and liquid phase. So for gaseous phase we give one color and for the liquid we give the other color. Here I have shown you with typical situation. Let us say this is a gaseous phase at the top of this one explain by red colored particles and over here this is the liquid phase explained by the blue colored particle. So definitely the interface will be lying somewhere over here which is differentiating between red and blue colored particles okay.

(Refer Slide Time: 17:18)



For two phase simulations:

Addition of surface tension force, $F_s = f_s \delta_s$

where, $f_s = -\sigma \nabla^2 C_s \frac{n}{|n|}$

and σ is the surface tension coefficient between the phases

Also, $C_{s,liquid} = 1$ $C_{s,gas} = 0$

and $n = \nabla C_s$

So what we do in numerical simulation of SPH also. Along with the other properties like Rho i and vi and ei, we define another property called ci okay which is called the color. So what we do, we give the color to the liquids as 1 and we give the color should gas as 0 okay. Now once we know the colors of the fluids what we can do, we can find out what is the normal direction of the interface because always we know the change of color if we try to find out the gradient of that one with respect to space that will be giving us the normal direction of the interface.

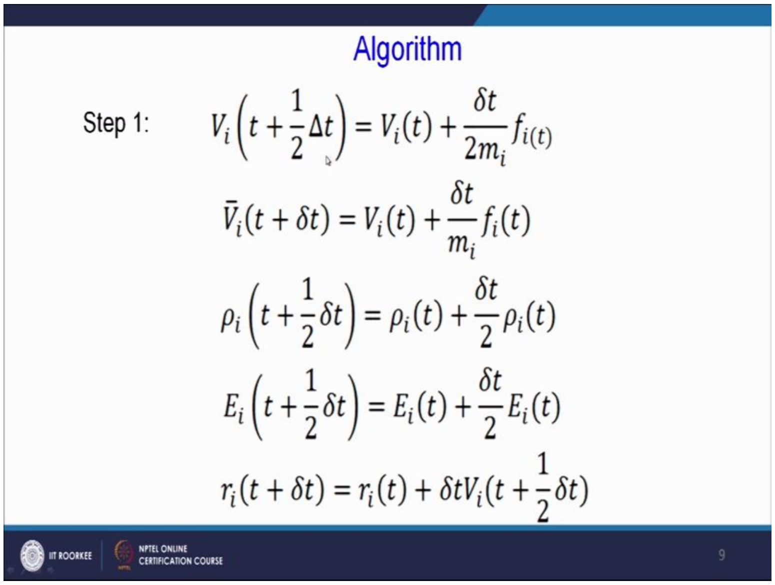
So n is nothing but del of Cs, where del of Cs is the color field okay. Once we know the value of n then easily we can find out the surface tension force okay surface tension force will be only applicable in the zone of the interface. So what we do, we use delta direct function okay to apply that surface tension force. So we can write down surface tension force fs will be actually fs which is surface tension force multiplied by this delta Dirac function delta s.

That means the delta Dirac function is only applicable near the interface that means delta s = 1 at the interface and elsewhere it is 0 okay. So what is this nature of fs. So fs can be written as – sigma, where sigma is the surface tension force into delta square Cs okay n /n bar. So, n /n bar is giving with the unit vector of the normal direction okay. Now, how to get once again this derivative of the Cs double derivative and single derivative over here. I have already shown you how to perform the derivatives on a field using f.

So here you will be doing the same approximations over Cs okay. Once I know all the particle Cs, we can find out its first derivative, second derivative using Kernel approximation okay. So in that way the capital Fs will be calculated and with each particle that portion will be added okay. Here I have shown you what is small fi. So this small fi will be coming into picture along with this capital Fs to find out what is the force on each particle right.

So in this way we go for 2 phase simulation. Let me show you the algorithm for time progress. So time advancement, so the method we follow is actually verlet algorithm. So what is algorithm let us say at present time step which is t th step we know all the variable that means starting from v Rho and e we know okay. So how to proceed it further so using vi(t) which is the which is the velocity t th time level, we find out some intermediate time table which is nothing but t + delt t / 2 okay some half time level we can say okay.

(Refer Slide Time: 20:15)



Algorithm

Step 1: $$V_i\left(t+\frac{1}{2}\Delta t\right) = V_i(t) + \frac{\delta t}{2m_i} f_{i(t)}$$

$$\bar{V}_i(t+\delta t) = V_i(t) + \frac{\delta t}{m_i} f_i(t)$$

$$\rho_i\left(t+\frac{1}{2}\delta t\right) = \rho_i(t) + \frac{\delta t}{2}\rho_i(t)$$

$$E_i\left(t+\frac{1}{2}\delta t\right) = E_i(t) + \frac{\delta t}{2}E_i(t)$$

$$r_i(t+\delta t) = r_i(t) + \delta t V_i(t+\frac{1}{2}\delta t)$$

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
9

So that we find out using vi (t +del t / 2) = vi (t) + del t / 2mi fi (t) okay. So you see over here we this fit is nothing but the force okay, force on the ith particle okay. At t th time step. So in the right hand side everything is known to me. So mass of each particle velocity at t th time and force at t th time we know and del t we will be choosing so time step we will be choosing. So depending on this will be finding out the velocity at some intermediate time step t+ del t / 2.

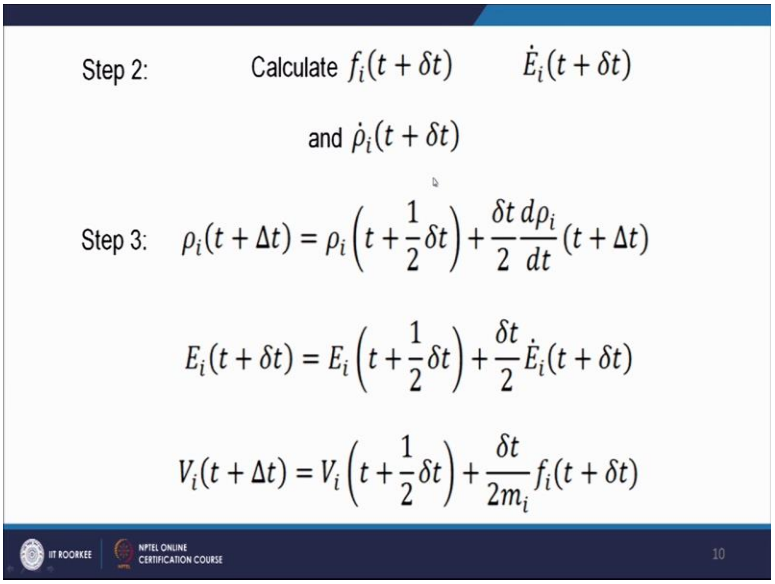
We can update this one or we can find out a new variable which is nothing but v i bar, remember this vi bar is not v (t) + delta t. So what we are finding out, some new variable we are defining Verlet algorithm. So vi bar at t +delta t, we find out based on the full time step now. Previous equation I have showed you for the half time step t + delta t /2 here we go for the full time step.

So you see the equation looks like similar but only thing is that here we had delta t/ 2, here we had only delta t because this is for the full time step. So we have found out 2 ways. One is vi at half time step another one is vi bar at full time step okay. Similarly we can obtain other variables like Rho i and ei for half time step.

So these equations will be similar to the first equation whatever I have showed dependent on the t th time step values and corresponding you know delta t values okay. Then we can also know update the value of r which is the position variable. so ri t + delta t = ri + delta t. So this time we require a velocity. So this velocity will be using as a half time step velocity. So v (t + delta t /2), here you see we have multiplied vi (t) + delta / 2.

So this ri (t) + delta t is actually found out by half time step okay. Next step, second step after finding out all these variables at the t+ delta t /2, 2th time step and a sudo velocity at t +delta tth time step, we will be going for calculation of actual value of force, energy and your density change okay.

(Refer Slide Time: 22:54)



Step 2: Calculate $f_i(t+\delta t)$ $\dot{E}_i(t+\delta t)$

and $\dot{\rho}_i(t+\delta t)$

Step 3: $$\rho_i(t+\Delta t)=\rho_i\left(t+\frac{1}{2}\delta t\right)+\frac{\delta t}{2}\frac{d\rho_i}{dt}(t+\Delta t)$$

$$E_i(t+\delta t)=E_i\left(t+\frac{1}{2}\delta t\right)+\frac{\delta t}{2}\dot{E}_i(t+\delta t)$$

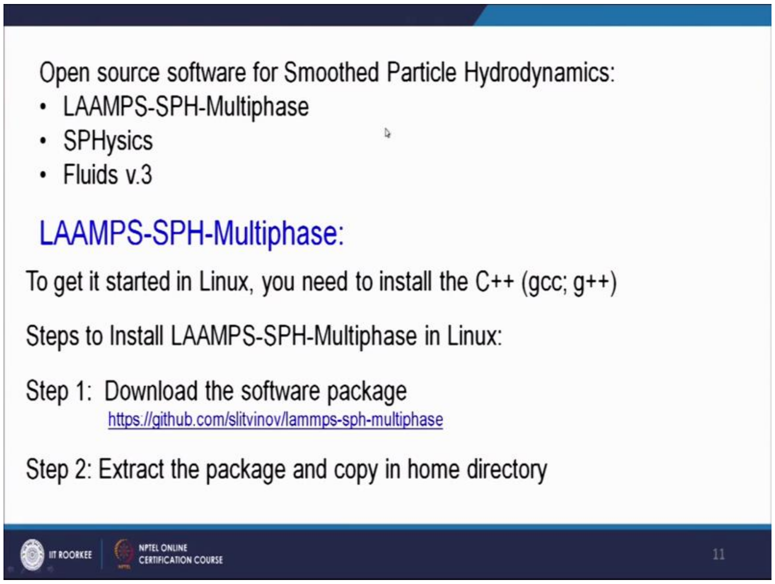
$$V_i(t+\Delta t)=V_i\left(t+\frac{1}{2}\delta t\right)+\frac{\delta t}{2m_i}f_i(t+\delta t)$$

So leaving but these are dot quantities with respect to time how much it is changing that we are trying to find out okay. Now these we can find out by putting the value of vi (t + delta t). So vi bar (t+ delta t) in your continuity momentum and energy equations. So if you remember the continuity momentum and energy equations we are having they are time derivatives in the left hand side okay.

So all these quantities can be found out using your momentum energy and continuity equations respectively okay. And using v bar delta t in the right hand side okay. So once you find out these updated values of fi ei dot and Rho dot, we will be updating the values of actual Rho ei and vi using the half time step values and the rates okay. You see Rho i (t + delta t) = Rho i (t +delta t / 2) +delta t /2 d Rho i /d, which is nothing but Rho i dot at (t + delta t), which we have found out in the second step okay.

Similarly ei can be written and vi can be also written. So once you get the new time step density energy and velocities values, we will forwarding merge for the (ti +2 delta t) okay. So in this way Verlet algorithm works okay. Next, let us see what are the freeware available to solve Smoothed Particle Hydrodynamics problems okay.

(Refer Slide Time 24:37)



Open source software for Smoothed Particle Hydrodynamics:

- LAAMPS-SPH-Multiphase
- SPHysics
- Fluids v.3

LAAMPS-SPH-Multiphase:

To get it started in Linux, you need to install the C++ (gcc; g++)

Steps to Install LAAMPS-SPH-Multiphase in Linux:

Step 1: Download the software package
https://github.com/slitvinov/lammps-sph-multiphase

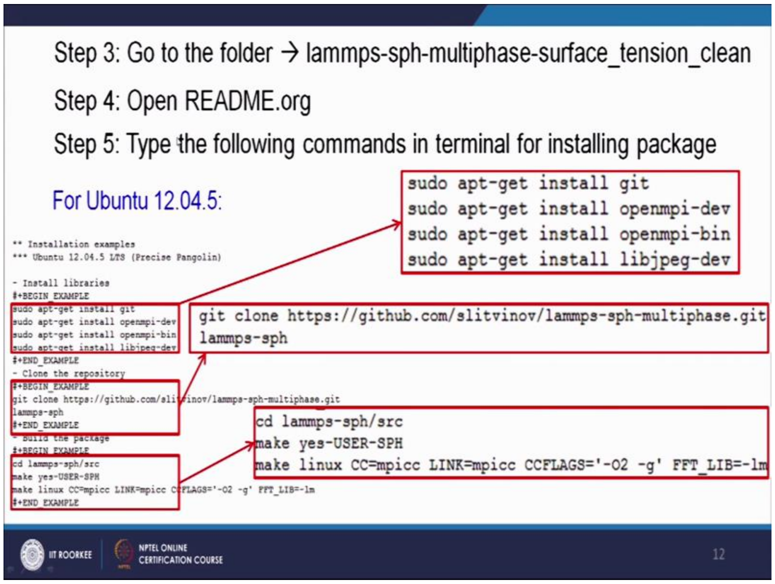
Step 2: Extract the package and copy in home directory

Usually we are having many open source software but here I will be giving you names as LAAMPS-SPH-Multiphase, SPHysics, Fluids v.3. So these are different software, where SPH is actually included as a no case okay. But today I will be showing you one simulation for a case study using LAAMPS-SPH-Multiphase okay. So first let me see what will be require to install this LAAMPS-SPH-Multiphase.

So we require a Unix platform okay also you require a working library of c ++, gcc or g ++ okay. What are the steps for installing the LAAMPS-SPH-Multiphase? First you need to download it from the software package, package id I have given you over here okay. And then you have to extract the package and copy it in the home directory. How to extract already I have shown you. So you can follow that step okay.

Then you have to go to the folder where you have extracted more or less the folder taking this name laamps-sph-multiphase surface tension clean okay. So you have to, so the main folder it will be laamps-sph-multiphase surface tension laamps-sph-multiphase. In that you will be finding out one folder called surface tension and another folder called clean okay. So you enter in the clean folder by doing cd cd cd.

(Refer Slide Time: 25:36)



Step 3: Go to the folder → lammps-sph-multiphase-surface_tension_clean

Step 4: Open README.org

Step 5: Type the following commands in terminal for installing package

For Ubuntu 12.04.5:

```
** Installation examples
*** Ubuntu 12.04.5 LTS (Precise Pangolin)

- Install libraries
#+BEGIN_EXAMPLE
sudo apt-get install git
sudo apt-get install openmpi-dev
sudo apt-get install openmpi-bin
sudo apt-get install libjpeg-dev
#+END_EXAMPLE
- Clone the repository
#+BEGIN_EXAMPLE
git clone https://github.com/slitvinov/lammps-sph-multiphase.git
lammps-sph
#+END_EXAMPLE
- Build the package
#+BEGIN_EXAMPLE
cd lammps-sph/src
make yes-USER-SPH
make linux CC=mpicc LINK=mpicc CCFLAGS='-O2 -g' FFT_LIB=-lm
#+END_EXAMPLE
```

```
sudo apt-get install git
sudo apt-get install openmpi-dev
sudo apt-get install openmpi-bin
sudo apt-get install libjpeg-dev
```

```
git clone https://github.com/slitvinov/lammps-sph-multiphase.git
lammps-sph
```

```
cd lammps-sph/src
make yes-USER-SPH
make linux CC=mpicc LINK=mpicc CCFLAGS='-O2 -g' FFT_LIB=-lm
```

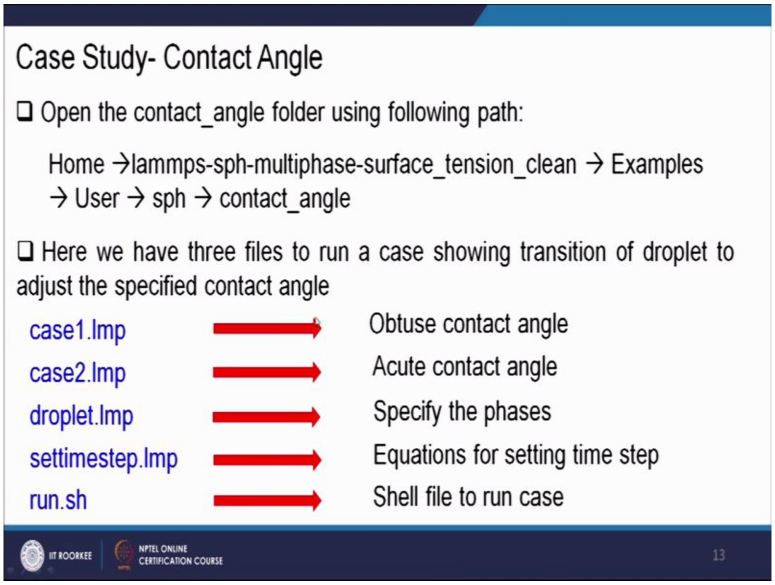
So already I have shown you how to do that and then open this read me.org okay .Then in that you type the following command in terminal for installing the packages okay. Usually I have shown here all these commands based on ubuntu 12. 4. 5 version okay. So what are the commands you need to give. So the major commands are this one. So you have to install these packages git openmpi dev, openmpi bin and libjpeg dev.

So these are necessary for running of LAAMPS-SPH-Multiphase okay. So you need to do sudo apt get install and the package name for all these packages. Then once this git is properly installed you can git clone from the library where the SPH multiphase is installed, the library is actually in slitvinov / laamps-sph-multiphase okay. So the address is written over here. So from that you can clone down using git. Okay then you can go to that folder where it has been actually downloaded that folder will be laamps-sph/src.

So that is the source folder, where all the codes are saved. You can go over there and write down make yes-USER-SPH. So that enables all the sph packages to run and you can use your SPH codes over here. Then you have to install the package using make linux CCmpicc. So this is one C++ directory and you have to link it with mpicc ccflags - 2 - g fft _ lib = - lm okay so you give this command, your installation will be done properly.

Let us now show you that how you can take a case study and you can solve using this LAAMPS-SPH-Multiphase. We will be showing you over here, a drop on a solid surface and due to contact angle how that is taking different shapes okay. So we have named that case study as contact angle. So case study is contact angle. So what you can do, you can open a folder okay called contact angle or if it is already there, you can go in the SPH-LAAMPS multiphase.

(Refer Slide Time: 28:11)



Case Study- Contact Angle

❑ Open the contact_angle folder using following path:

Home →lammps-sph-multiphase-surface_tension_clean → Examples → User → sph → contact_angle

❑ Here we have three files to run a case showing transition of droplet to adjust the specified contact angle

case1.lmp	→	Obtuse contact angle
case2.lmp	→	Acute contact angle
droplet.lmp	→	Specify the phases
settimestep.lmp	→	Equations for setting time step
run.sh	→	Shell file to run case

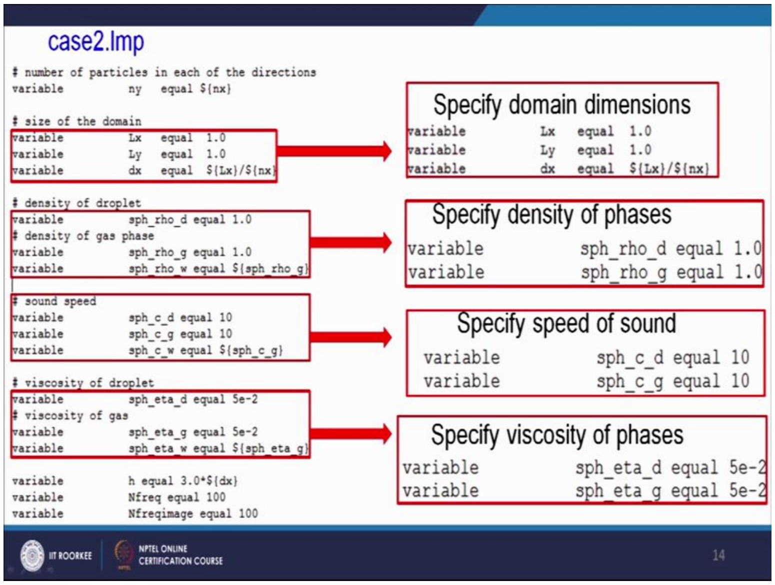
IIT ROORKEE NPTEL ONLINE CERTIFICATION COURSE 13

So it is already this is a case study or already developed case study in LAAMPS-SPH-Multiphase. So there from we have taken so it is already existing folder. If you are having some new case you have to develop the folder and then you need to write down. So the case study folder we have to go which is nothing but Home->laamps->sph-multiphase-surface_tension_clean there you can go to examples User-> sph-> contact_angle.

So this is the path of the directory where you need to go. In this path contact angle, in this folder contact angle you can find out 5 files okay. In which 2 files case1 lmp and case2 lmp those are actually for basic information but you need to run this droplet lmp, settimestep.lmp and run.sh okay for setting up the simulations okay. So let me explain one by one what all these files are dealing with.

So case1 lmp and case2 lmp these are giving you the definition of obtuse and acute contact angles. So you know that contact angle can be more than and less than 90 degree depending on that having cases of obtuse and acute contact angles okay. I will be showing you details of 1.lmp file. So case2 lmp I will be showing you and I will be giving you where this case1 lmp is differing rest commands will be same.

Apart from that I will be showing the placement of this droplet.lmp that means initial placement of the drop that is being specified in this file. For running the calculations we need to set the time. So settimestep.lmp will be giving that 1 and for running the informations required in run.sh. Let us see quickly what is case2 lmp important things I have highlighted over here. You see this part is given over here this lx, ly and dx these are actually your domain span in x direction and y direction and dx is actually your smallest size of the cell okay, which will be pressing the particles initially.
Initially the particles are placed in a rectangular grid.
(Refer Slide Time: 30:13)



case2.lmp

```
# number of particles in each of the directions
variable            ny   equal ${nx}

# size of the domain
variable            Lx   equal  1.0
variable            Ly   equal  1.0
variable            dx   equal  ${Lx}/${nx}

# density of droplet
variable            sph_rho_d equal 1.0
# density of gas phase
variable            sph_rho_g equal 1.0
variable            sph_rho_w equal ${sph_rho_g}

# sound speed
variable            sph_c_d equal 10
variable            sph_c_g equal 10
variable            sph_c_w equal ${sph_c_g}

# viscosity of droplet
variable            sph_eta_d equal 5e-2
# viscosity of gas
variable            sph_eta_g equal 5e-2
variable            sph_eta_w equal ${sph_eta_g}

variable            h equal 3.0*${dx}
variable            Nfreq equal 100
variable            Nfreqimage equal 100
```

Specify domain dimensions

```
variable            Lx   equal  1.0
variable            Ly   equal  1.0
variable            dx   equal  ${Lx}/${nx}
```

Specify density of phases

```
variable            sph_rho_d equal 1.0
variable            sph_rho_g equal 1.0
```

Specify speed of sound

```
variable            sph_c_d equal 10
variable            sph_c_g equal 10
```

Specify viscosity of phases

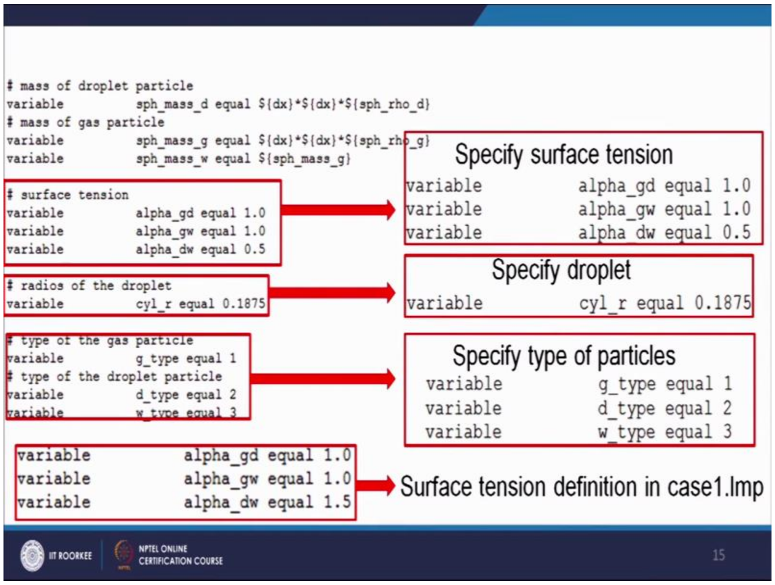
```
variable            sph_eta_d equal 5e-2
variable            sph_eta_g equal 5e-2
```

So that is the grid spacing okay. So we have taken over here lx /nx. nx is the predefined value which we give in the file while running I will be showing you where lx value will give here. Similarly we have given the density of the droplet you see, we are having 2 things over here sph_ rho_d and sph_rho_g. So initially we have given these values as 1 okay. Similarly here we are having speed of sound okay sph_c_d.

So C means speed of sound because we have seen that speed of sound is very important at least for von Newman-Richter artificial viscosity. So we have given those values as 10, 10 okay. Here we have given the viscosity of the droplet you see the viscosity is written as eta okay. So eta d and eta g so initially we have given those values as 5 * e to the power – 2.

Then if you go further we will be finding out in this position we have given the surface tension. As we know that to create the contact angle gradient contact angle we need to give 3 surface tensions sigma s w, sigma l w and sigma l s. So liquid gas, liquid wall and liquid solid wall. So those values are written over here. So alpha is actually your surface tension over here. So alpha gd between gas and your droplet here this gas and wall and here droplet and wall okay.

(Refer Slide Time 31:50)



```
# mass of droplet particle
variable          sph_mass_d equal ${dx}*${dx}*${sph_rho_d}
# mass of gas particle
variable          sph_mass_g equal ${dx}*${dx}*${sph_rho_g}
variable          sph_mass_w equal ${sph_mass_g}
```

```
# surface tension
variable          alpha_gd equal 1.0
variable          alpha_gw equal 1.0
variable          alpha_dw equal 0.5
```

Specify surface tension

```
variable          alpha_gd equal 1.0
variable          alpha_gw equal 1.0
variable          alpha_dw equal 0.5
```

```
# radios of the droplet
variable          cyl_r equal 0.1875
```

Specify droplet

```
variable          cyl_r equal 0.1875
```

```
# type of the gas particle
variable          g_type equal 1
# type of the droplet particle
variable          d_type equal 2
variable          w_type equal 3
```

Specify type of particles

```
variable          g_type equal 1
variable          d_type equal 2
variable          w_type equal 3
```

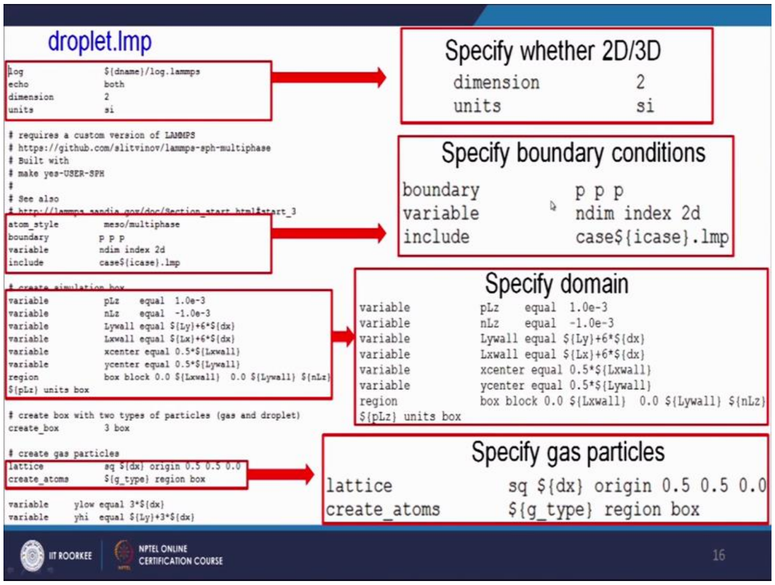
```
variable          alpha_gd equal 1.0
variable          alpha_gw equal 1.0
variable          alpha_dw equal 1.5
```

Surface tension definition in case1.lmp

So these values are actually used in our Young's equations we find out over there using this we can actually evaluate the value of the contact angle okay. Then here we have given the radius of the droplet you see the cylindrical radius of the droplet we have given as 0. 1875 okay. Then here in this equation what we have given the type of particles. We are having actually 3 different types of particles first type of particle g type = 1.

We have given for the gas okay around the droplet, d type = 2 that we have given for the droplet inside the particles of the droplet and then w type the wall type particles we have given as 3 okay. Now this is more or less the file what we see in case of your case2.lmp okay. case2.lmp is actually for acute contact angle okay.

So this combinations of alphas will be giving you acute contact angle okay. But if you see this case1 dot lmp these values are actually modified to 1, 1 and 1. 5. See over here dw value is actually taking important role for deciding the obtuse or acute. Next let us see the droplet positions so the droplet.lmp. So at the beginning we have shown to how to save that dimension and the units. So dimension is 2 and the unit is that not much important in sph-laamps.
(Refer Slide Time: 33:50)



droplet.lmp

```
log                 ${dname}/log.lammps
echo                both
dimension           2
units               si

# requires a custom version of LAMMPS
# https://github.com/slitvinov/lammps-sph-multiphase
# Built with
# make yes-USER-SPH
#
# See also
# http://lammps.sandia.gov/doc/Section_start.html#start_3
atom_style          meso/multiphase
boundary           p p p
variable           ndim index 2d
include            case${icase}.lmp

# create simulation box
variable            pLz    equal  1.0e-3
variable            nLz    equal  -1.0e-3
variable            Lywall equal ${Ly}+6*${dx}
variable            Lxwall equal ${Lx}+6*${dx}
variable            xcenter equal 0.5*${Lxwall}
variable            ycenter equal 0.5*${Lywall}
region              box block 0.0 ${Lxwall}  0.0 ${Lywall} ${nLz}
${pLz} units box

# create box with two types of particles (gas and droplet)
create_box          3 box

# create gas particles
lattice             sq ${dx} origin 0.5 0.5 0.0
create_atoms        ${g_type} region box

variable     ylow equal 3*${dx}
variable     yhi  equal ${Ly}+3*${dx}
```

Specify whether 2D/3D

```
dimension           2
units               si
```

Specify boundary conditions

```
boundary           p p p
variable           ndim index 2d
include            case${icase}.lmp
```

Specify domain

```
variable     pLz    equal  1.0e-3
variable     nLz    equal  -1.0e-3
variable     Lywall equal ${Ly}+6*${dx}
variable     Lxwall equal ${Lx}+6*${dx}
variable     xcenter equal 0.5*${Lxwall}
variable     ycenter equal 0.5*${Lywall}
region       box block 0.0 ${Lxwall}  0.0 ${Lywall} ${nLz}
${pLz} units box
```

Specify gas particles

```
lattice          sq ${dx} origin 0.5 0.5 0.0
create_atoms     ${g_type} region box
```

We always use it as non dimensional but you have to give si for the sake of completion of the codes okay. Over here you see in this we have specified the boundary conditions. So boundary we have given as p p p that mean every where we are having the boundary as reflective boundary conditions okay. The variables that we have defined over here okay ndim index and 2d so these decides that you are having 2-dimensional domain okay.

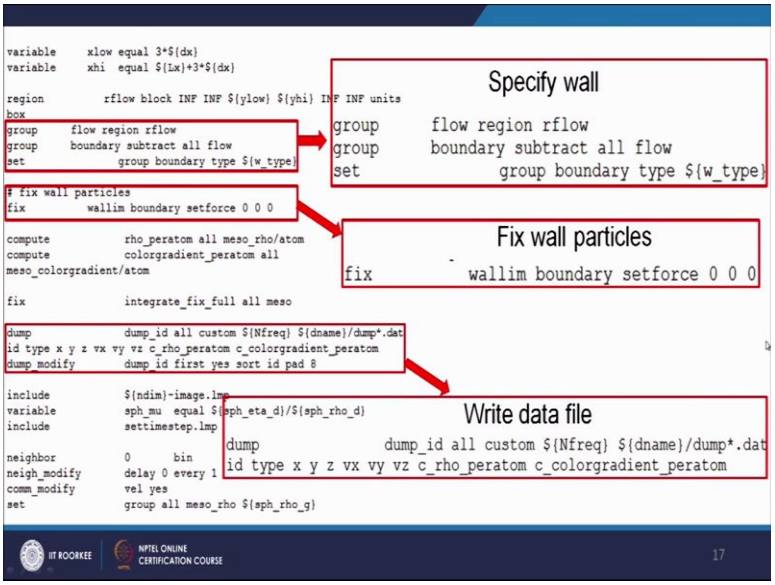
And here you see, we are including which case already I have shown you case1 and case2.lmp for obtuse and acute angle. So here I can fill it by giving the proper value of i case which obtuse and acute that we can decide over here. Now in this we have defined the domain if you see here we have given first the z limit. As it is a 2d problem in the z limit we have given you a very small thickness.

So which is nothing but starting form -1 into 10 to the power -3 to + 1 into 10 to the power -3. And we have also selected a small domain for wall in the y direction and x direction. So you see over here ly wall I have actually extended the y portion little bit of 6 times of the dx. So dx is a small unit which I have already shown you. 6 times of that we have actually expand with the domain in y direction just to increase the domain.

Similarly in lx wall is the also similar lx + 6 * dx. x center and y center those are actually the center location of the drop. So they are we have given 0. 5*lx wall and 0. 5*ly wall okay. lx wall and ly wall we have already defined. Then let us create a region with the box which is a 2 dimensional box starting from 0 to lx wall 0 to ly wall and in the z direction starting from nlz to plz as it is a 2 dimensional problem so that it will be very small.

Now here we have specified the gas particles you see first we have specified all the gas particles inside the domain that means everywhere with a square box we have decided the Cartesian grids, in the grids we at the center of the grid 0. 5 and 0. 5 we have actually placed a particle okay.

(Refer Slide Time: 36:31)



```
variable     xlow equal 3*${dx}
variable     xhi  equal ${Lx}+3*${dx}

region          rflow block INF INF ${ylow} ${yhi} INF INF units
box
group      flow region rflow
group      boundary subtract all flow
set                 group boundary type ${w_type}

# fix wall particles
fix          wallim boundary setforce 0 0 0

compute              rho_peratom all meso_rho/atom
compute              colorgradient_peratom all
meso_colorgradient/atom

fix                  integrate_fix_full all meso

dump                 dump_id all custom ${Nfreq} ${dname}/dump*.dat
id type x y z vx vy vz c_rho_peratom c_colorgradient_peratom
dump_modify          dump_id first yes sort id pad 8

include              ${ndim}-image.lmp
variable             sph_mu  equal ${sph_eta_d}/${sph_rho_d}
include              settimestep.lmp

neighbor             0       bin
neigh_modify         delay 0 every 1
comm_modify          vel yes
set                  group all meso_rho ${sph_rho_g}
```

Specify wall

```
group      flow region rflow
group      boundary subtract all flow
set                 group boundary type ${w_type}
```

Fix wall particles

```
fix          wallim boundary setforce 0 0 0
```

Write data file

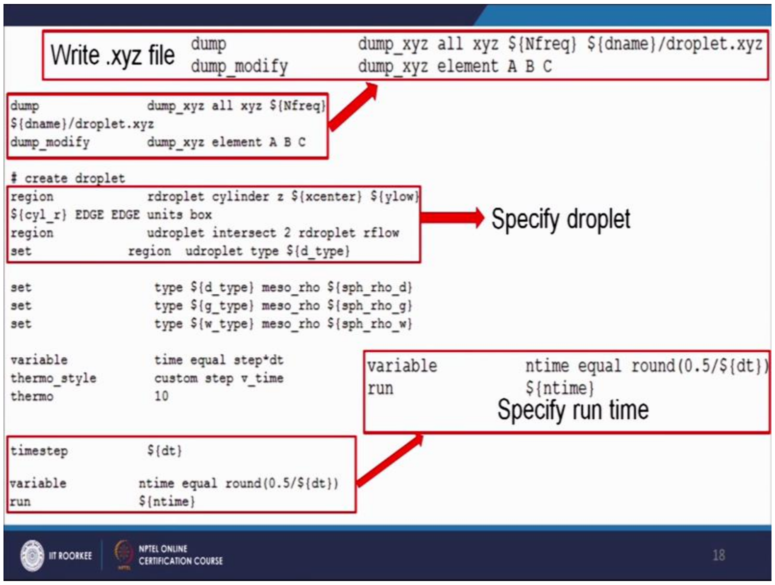
```
dump                 dump_id all custom ${Nfreq} ${dname}/dump*.dat
id type x y z vx vy vz c_rho_peratom c_colorgradient_peratom
```

So initially we are having a rectangular discretization of the particles. Then over here in this we have given the wall type because at the wall the droplet is resting so we have actually said out specified out the wall. So first we have created a group which is for the flow direction okay. So first the flow direction means where the droplet and the gaseous phase is kept okay.

Then what we have done we have subtracted the boundary okay from the flow. So we have created a section which is actually the boundary section okay. So from the overall region if you subtract our flow region, r flow region then we will be getting the boundary okay. Then we have set the boundary okay set the boundary as a wall type. So w type we have given over here okay.

Then we have given the force to these wall particles which is nothing but 0 0 0 that means we have make the wall particles as non moving particles okay because those particles cannot move. Now whenever you will be running the simulation it will be writing down the value so for writing the value over here you see we are using dump command. Dump command is useful for writing the value and we are writing all the values starting from id of the particle those are the i's or j's.

(Refer Slide Time: 37:36)



Write .xyz file

```
dump                dump_xyz all xyz ${Nfreq} ${dname}/droplet.xyz
dump_modify         dump_xyz element A B C
```

```
dump                dump_xyz all xyz ${Nfreq}
${dname}/droplet.xyz
dump_modify         dump_xyz element A B C

# create droplet
region              rdroplet cylinder z ${xcenter} ${ylow}
${cyl_r} EDGE EDGE units box
region              udroplet intersect 2 rdroplet rflow
set              region  udroplet type ${d_type}
```

Specify droplet

```
set                 type ${d_type} meso_rho ${sph_rho_d}
set                 type ${g_type} meso_rho ${sph_rho_g}
set                 type ${w_type} meso_rho ${sph_rho_w}

variable            time equal step*dt
thermo_style        custom step v_time
thermo              10

timestep            ${dt}

variable           ntime equal round(0.5/${dt})
run                ${ntime}
```

```
variable            ntime equal round(0.5/${dt})
run                 ${ntime}
```

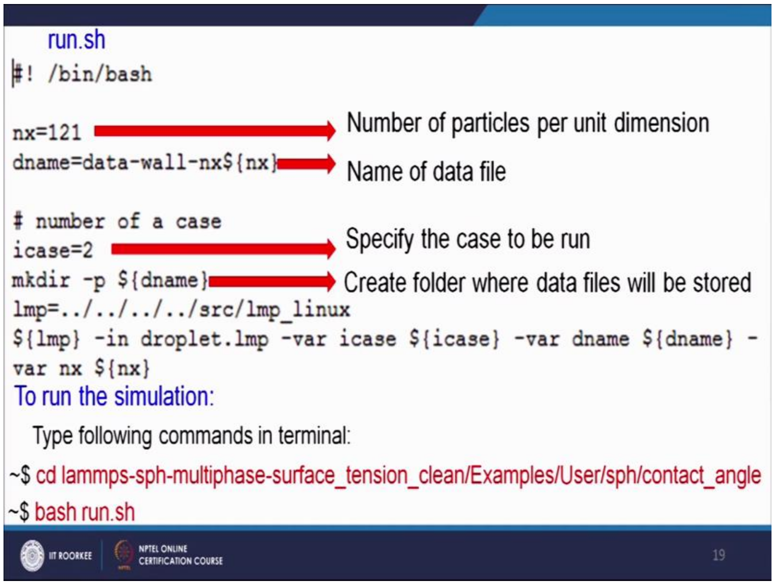
Specify run time

And then type of the particle 1,2 or 3 for the gas liquid and wall respectively x locations, y locations, z locations their respective velocities and this is you viscosity color gradient so things you have to write down okay. Then we will be showing you over here that how using dump command we can go for you know velocity dumping and for your values dumping in a separate file which can be reading some external software like vmd okay.

So here this is the command for dump for all the particles x, y and z locations we are writing okay all the elements x,y, z locations we are writing over here okay. Here we have specified the droplet because you see we have just created the center we have not shown the droplet is having which type of particle.

So here you see we have first shown that what is the droplet over here starting having a cylinder okay at x center and ylow okay, cylinder radius given as rdroplet okay and then we have given that cylinder inside the cylinder whatever particles you are having that is the droplet type okay that is 2 type okay. Then here we are running the simulation with a time step. So here we are have calculated the time step. So time step = 0. 5 / delta t okay.

(Refer Slide Time: 38:58)



run.sh

```
#! /bin/bash

nx=121
dname=data-wall-nx${nx}

# number of a case
icase=2
mkdir -p ${dname}
lmp=../../../../src/lmp_linux
${lmp} -in droplet.lmp -var icase ${icase} -var dname ${dname} -
var nx ${nx}
```

Number of particles per unit dimension

Name of data file

Specify the case to be run

Create folder where data files will be stored

To run the simulation:

Type following commands in terminal:

```
~$ cd lammps-sph-multiphase-surface_tension_clean/Examples/User/sph/contact_angle
~$ bash run.sh
```

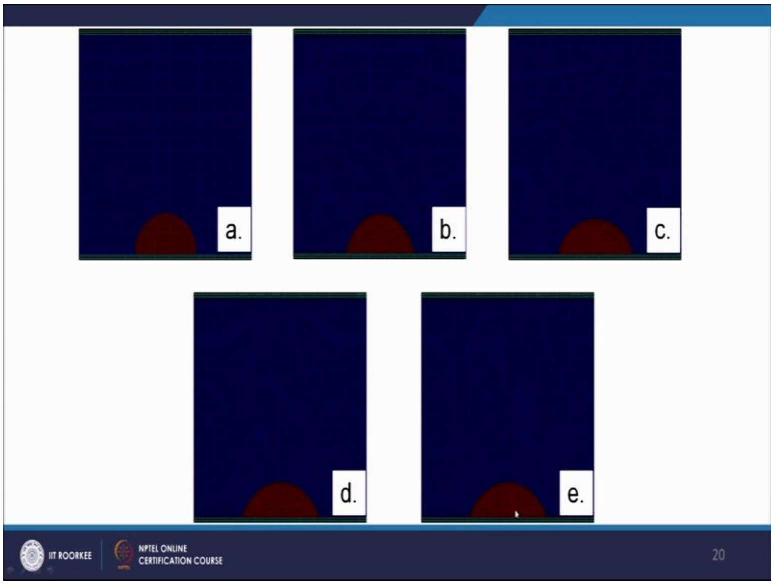
So in this way the time step delta t has been selected and the code will be running further okay. And it will be running for n times so number of time you can decide okay. Now this is the run file using which we can decide the numbers okay for running all those previous files. For example you see nx 121 is very important this will be deciding that what will be the dx value x, y and z directions okay.

Then here you see the name of the data file, where it will be writing the data file in which fashion that we have given over here. icase = 2 will be deciding whether to choose the obtuse or acute file okay. So icase was going in the lmp file okay from where we choosing that which one we need to take okay. Similarly to create a folder where will be storing the values the output files that will be right here okay.

So mkdir - p and then the name okay over here. Now first what we need to do, we need to run the simulations, for running the simulation we need to go for the respective folder so this is the command using cd we can go to the respective folder, the path is given over here and then finally just a single command bash run.sh will be running our code okay.

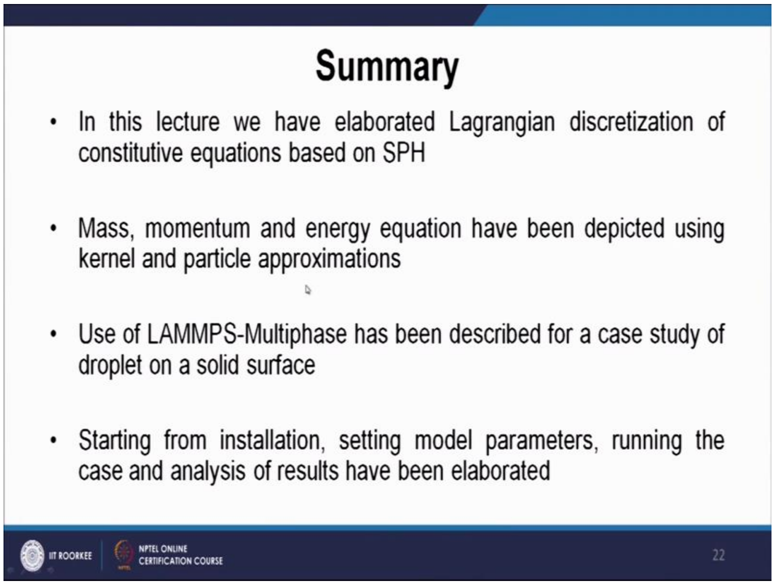
So that means this will be running this r run.sh file which will be giving the values to the corresponding lmp file for running okay. So this is the results for the case, here you can see we started with a hemispherical droplet due to contact angle okay it has taken flat end shape over here. So we should change the values of the alphas okay in the in the lmp files okay then we will be finding out this is becoming more flatting depending on the contact angle. A close look of this one you can see over here okay.

(Refer Slide Time: 40:16)



So at the end of this lecture we have elaborated the Lagrangian discretization of the constitute equations of the SPH, mass momentum energy equations have been elaborated okay, using LAMMPS- Multiphase we have shown you one case study, where droplet has been taken a shape over the solid surface. Starting from installation, setting model parameters, running the case analysis of results have been elaborated.

(Refer Slide Time : 40:36)

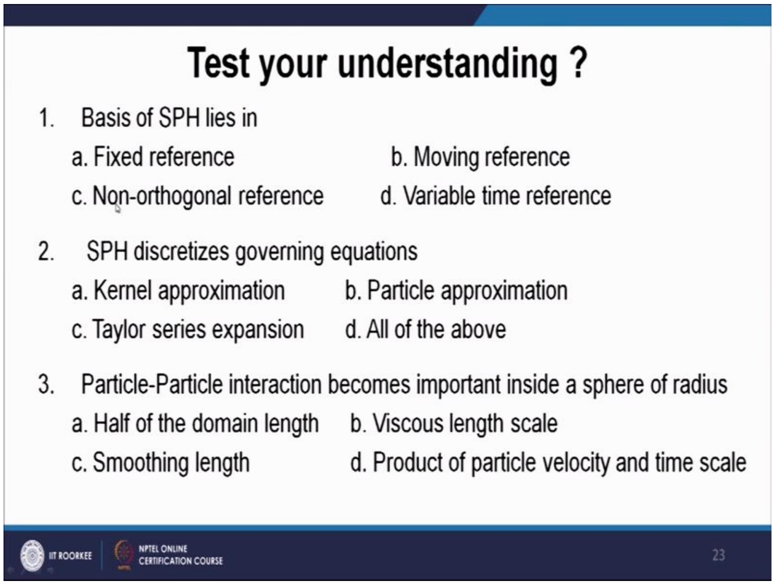


Summary

- In this lecture we have elaborated Lagrangian discretization of constitutive equations based on SPH
- Mass, momentum and energy equation have been depicted using kernel and particle approximations
- Use of LAMMPS-Multiphase has been described for a case study of droplet on a solid surface
- Starting from installation, setting model parameters, running the case and analysis of results have been elaborated

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE 22

(Refer Slide Time: 40:58)



Test your understanding ?

1. Basis of SPH lies in

   a. Fixed reference b. Moving reference

   c. Non-orthogonal reference d. Variable time reference

2. SPH discretizes governing equations

   a. Kernel approximation b. Particle approximation

   c. Taylor series expansion d. All of the above

3. Particle-Particle interaction becomes important inside a sphere of radius

   a. Half of the domain length b. Viscous length scale

   c. Smoothing length d. Product of particle velocity and time scale

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE 23

Now test your understanding we are having 3 questions basis of SPH lies in fixed reference, moving reference, non orthogonal reference and variable time reference. Correct answer is moving reference. So it is Lagrangian in nature. SPH discretizes governing equations in using Kernel approximation, Particle approximation, Taylor series expansion, are all of the above. Already I have shown you kernel approximation and placement of particles.

So this is actually correct answer is both kernel approximation and particle approximation. Third question Particle-Particle interaction becomes important inside a sphere of radius. 4 answers we are having. Half of the domain length, viscous length scale, smoothing length and product of particle velocity and time scale. So correct answer is obviously smoothing length I have shown you a figure also. So with this I end this lecture, thank you.