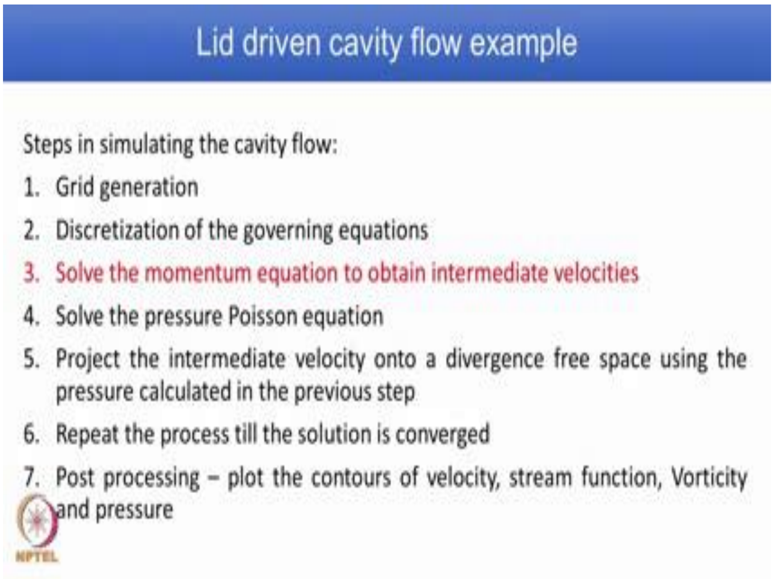
Foundation of Computational Fluid Dynamics
Dr. S. Vengadesan
Department of Applied Mechanics
Indian Institute of Technology, Madras

Lecture - 40

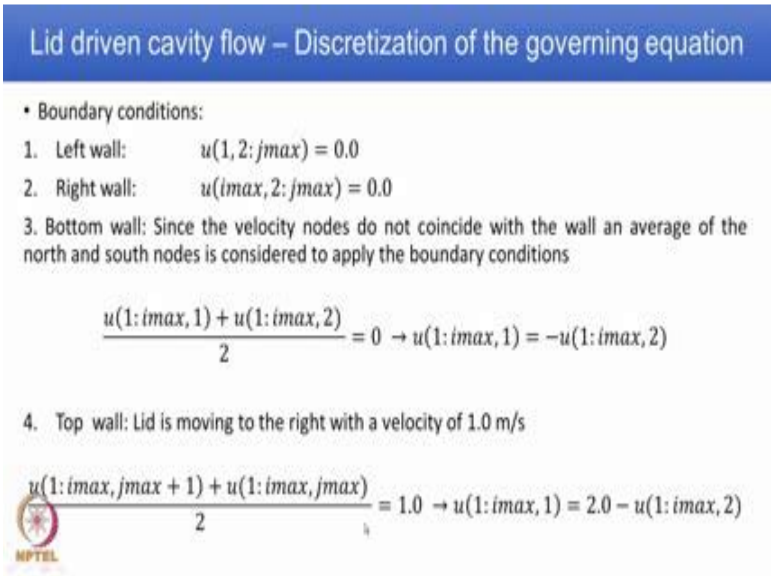
(Refer Slide Time: 00:42)



Greetings and it is my pleasure to welcome you to special course on CFD. In this week, we mention, we will have a demonstration of a working code, employing different numerical strategies we learned during the last seven weeks. The problem we consider to demonstrate the flow lid driven cavity. We listed in last module algorithm or steps involved in arriving at a code and numerical strategy. Step one grid generation then discretization of the governing equations, solving the momentum equation to obtain intermediate velocities, solving pressure Poisson equation then project the intermediate velocity onto a divergence free space, using the pressure obtained through pressure Poisson equation, and repeat the above steps until convergence is ensure. Once we obtained the results, then we look it to the flow through different post processing, and how to plot different quantities.

The last two modules we basically did how to do grid generation, because the geometry is very simple, we consider uniform grid and the structured grid, 4 by 4 mesh arrangements and then we started doing something about governing equation discretization. We started with u-momentum equation; in the u-momentum equation, we had diffusion term explained in the last module. In this module, we will particularly focus further on u-momentum equation, considering the convection term; and then extend this procedure for the next momentum equation that is v momentum equation.

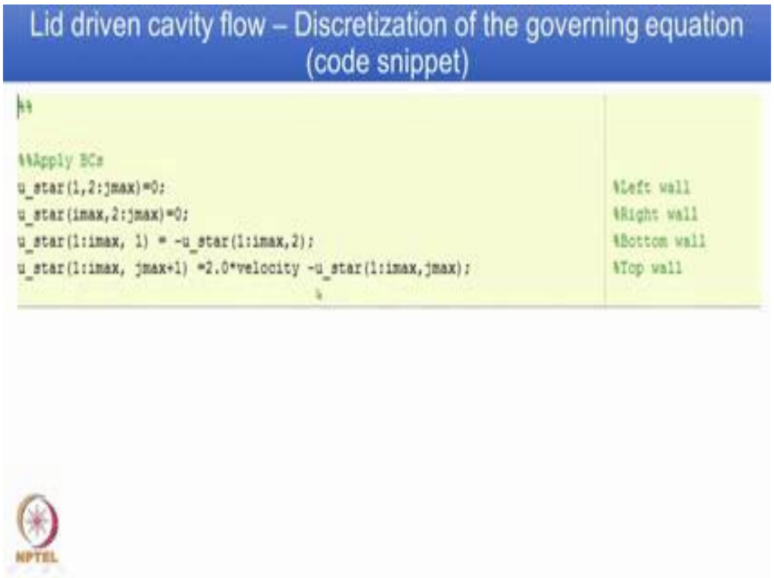
(Refer Slide Time: 02:29)



Continuation of the last class, whenever you solve momentum equation, we have to implement boundary conditions, and we defined because it is wall on three sides for the primary variable u and v, we define no slip condition. And on the top wall, we had velocity driving the lid as the boundary condition. And the pressure, we apply Neumann type of boundary condition that is derivatives of pressure equal to zero. So, first we see, how to implement boundary condition for u velocity. So, left wall u and for left wall i is actually 1, and j running from 2 to j max equal to 0.0. On the right wall, grid in x direction is i max, and again it is running in the j direction from 2 to j max equal to 0.0. Bottom wall since for u node, they do not coincide with the wall, we have to somehow ensure the no slip condition is also satisfied. So, we take average of north and south nodes, which indirectly will ensure the no slip condition is imposed on the bottom wall.

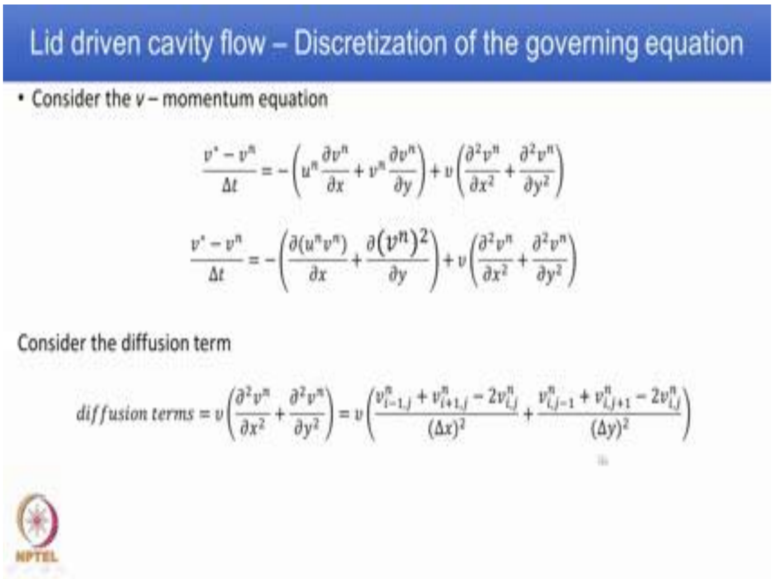
So, mathematically it is u again at 1 that i is equal to 1 i max to 1 plus u 1 and i max comma 2 divide by 2 is zero which will result in u 1 i max comma 1 equal to minus u 1 i max comma 2. So, this way you ensure appropriate boundary condition imposed on the bottom wall. On the top wall, we mentioned the lid is moving to the right with the velocity of 1 metre per second, and that is also ensure in the similar way that is take a average. So, in this case, now it is u 1 to i max; in j it is j max plus 1 plus u 1 to i max and j it is j max divided by 2 and that should be set to the desire velocity that is 1.0. Now you rearrange, you get expression u 1 to i max comma 1 equal to 2.0 minus u 1 to i max comma 2.

(Refer Slide Time: 05:28)



Now in terms of code, you see that u star because, we first do without pressure term included solve the u-momentum equation. So, such a velocity is given superscript u star v star that they are all predicted velocities then once you corrected, they will be set to that actual velocity. So, this is done immediately after solving u momentum equation without considering pressure term. So, it is called u star, so u star 1 comma 2 to j max in the j direction equal to 0. So, this is for enforcing boundary condition on the left wall; similarly the next line u star i max comma 2 to j max for j direction equal to 0, this is to impose boundary condition on the right wall, and similarly for bottom wall as well as for top wall.

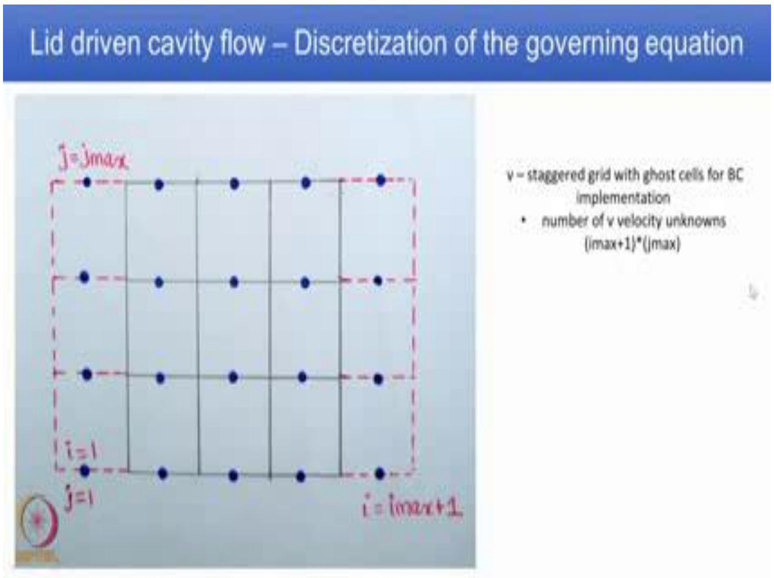
(Refer Slide Time: 06:37)



We have now explained in detail u-momentum equation considering separately diffusion term and convection term, and how to impose boundary condition. We will extend this procedure for the second momentum equation that is v-momentum equation, without pressure term in the v-momentum equation, equation is written as shown here, because we are solving without pressure term, the velocity is given superscript. So, it is v star minus v n by delta t equal to convection term from the left side is got to the right side. So, u n dou v n by dou x plus v n dou v n by dou y with a minus sign, because it is brought by the left side plus this viscous diffusion term as shown here. We mention we are solving explicitly, so all the superscript for other quantities are with n.

Last module we also mention the convection term can be rewritten as shown here. So, this u n is brought inside the partial derivatives as shown here that is dou by dou x of u n v n plus; again for the second term, v n is brought inside the square partial derivatives. So, dou do by dou y of v n square plus the diffusion term. In the next slide, I am going to explain how this can be written and what actually it results. Consider the diffusion term first. So, diffusion term that is nu dou square by dou x square of v n plus dou square by dou y square of v n at nth level, and that we are using second order central differences scheme. So, the first term is for the x direction, so we have i minus 1 comma j i plus 1 comma j minus 2 times v at i comma j evaluated at the nth level divide by delta x square. Now the second term is for the second derivative in the y direction that is evaluated again using central differences scheme in the second direction, so v at i j minus 1 v at i j plus 1 minus 2 times v at i comma j evaluated at nth level divided by delta y square.

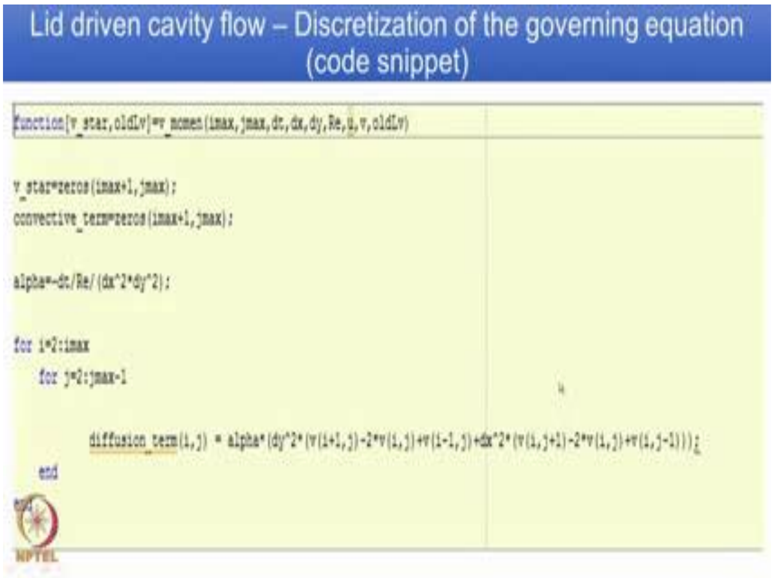
(Refer Slide time: 09:12)



We mention in order to have boundary conditions enforce we should have ghost cells that is they are not actually part to the grid that have you define, but there are extra nodes in particular directions. And for simplicity, we take those extra grid lines at the same distance as a original grid lines. For example, we defined 4 by 4 grid lines. So, we have four in vertical line, so 1, 2, 3, 4; similarly four in the horizontal, so 1, 2, 3, 4. Now for the v momentum equation, because we are following the standard grid arrangements of variables, we need to have one more extra grid line on either side as a ghost cell or ghost node. So, these are shown by a red colour line. So, this is in the left side and another one is in the right side as shown here.

Now for simplicity sake, we defined delta x of the ghost grid line and ghost cell same as the immediate adjacent cell. Similarly for the other direction, because v is in standard grid arrangements, they are all stored along the horizontal line and they are shown this picture with the full circle as shown here. So, this is for the v velocity, similarly for these are all storing v velocity. So, v standard grid with ghost cell for boundary condition implementation. We have had a similarity when we did u momentum equation, but it was in the j direction; now for v momentum equation, it is in the i direction. Because of this extra ghost grid or ghost cell number of v velocity unknowns now becomes i max plus 1 into j max.

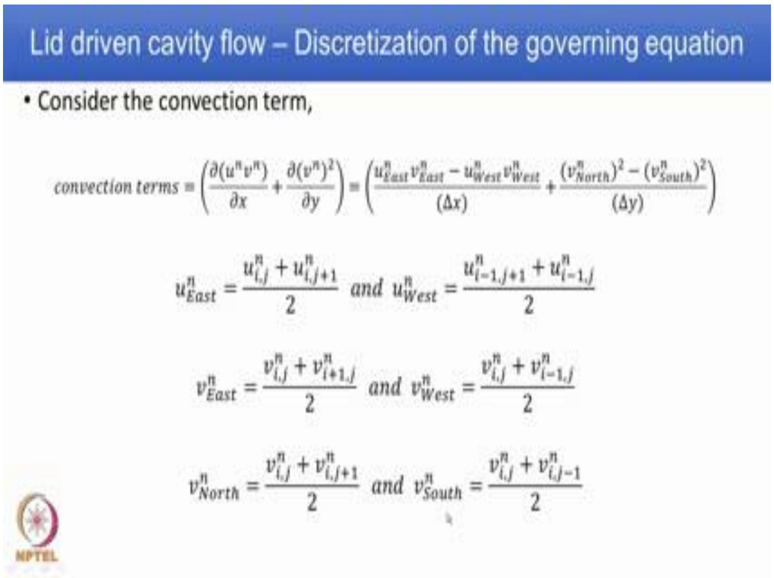
(Refer Slide Time: 11:17)



So, corresponding code part of the code snippet, so function v star old v equal to v momentum and you have argument defined as i max, j max, dt, dx, dy, Reynolds number u, v and old v. So, v star you defined memory zeros i max plus 1 j max; convective term zeros i max plus 1 j max and alpha is a new variable you defined. If you can recall we have unsteady term on the left side. So, v star minus v n by delta t equal to the convection term brought from the left side to the right side and then we have a diffusion term. The diffusion term dou square v n by dou x square plus dou square v n by dou y square. So, if you discretize, we have delta x square delta y square, and this alpha is taking those terms those coefficient appropriately. So, minus dt is brought from the left side and Reynolds number associated with nu, and delta x square and delta y square.

So, when you write the diffusion term, we have alpha, and for the first term that is dou square v by dou x square. We explain that we are doing second order central differences scheme then we have corresponding terms here that is v i plus 1 comma j v i comma j v i minus 1 comma j and this should be divided by delta x square because alpha has both delta x square and delta y square, we multiply by delta y square for the first step. Similarly the second term in the diffusion term is dou square v by dou y square; if you write it in difference form it is by delta y square, and we are multiplying by alpha commonly. So, extra term is adjusted in numerated as dx square into the second order central differences scheme for the second term, so we have v evaluated v taken from i j plus 1 minus 2 v i comma j plus 2 v i j minus 1.

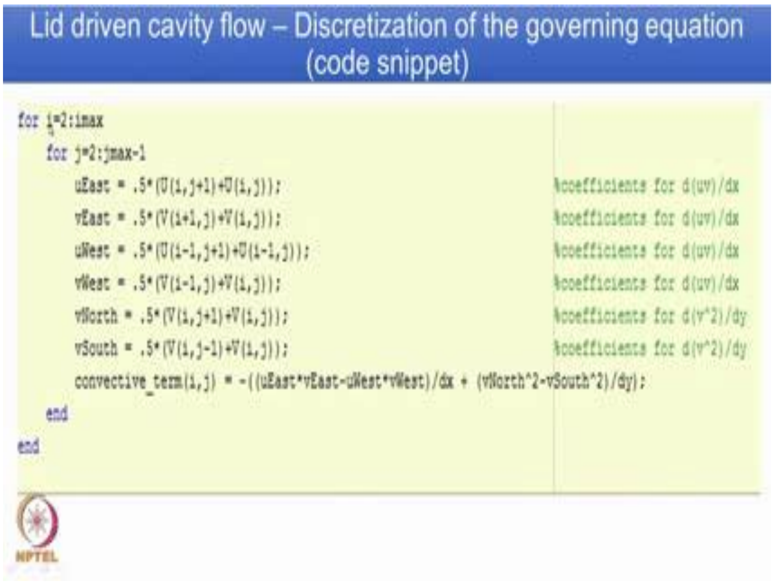
(Refer Slide Time: 13:48)



So, once we get diffusion term explained, we will now move on to convection term. So, convection term is rewritten including the u multiplying and v multiplying into the partial derivative. So, it is dou by dou x of u n v n plus dou by dou y of v n square, and we write in terms of north, east, west, south. So, we have u east at nth level; v east at nth level; minus u west at n th level, and v west at n th level divided by delta x. Similarly for the second term as v north from n time level square minus v south at n time level square divide by delta y. Now if you recall what we learned in week four lesson, convection term treatment, we have a different approximation procedure, central differential type of approximation, pure upwinding approximation, QUICK type of approximation, power law scheme, hybrid scheme.

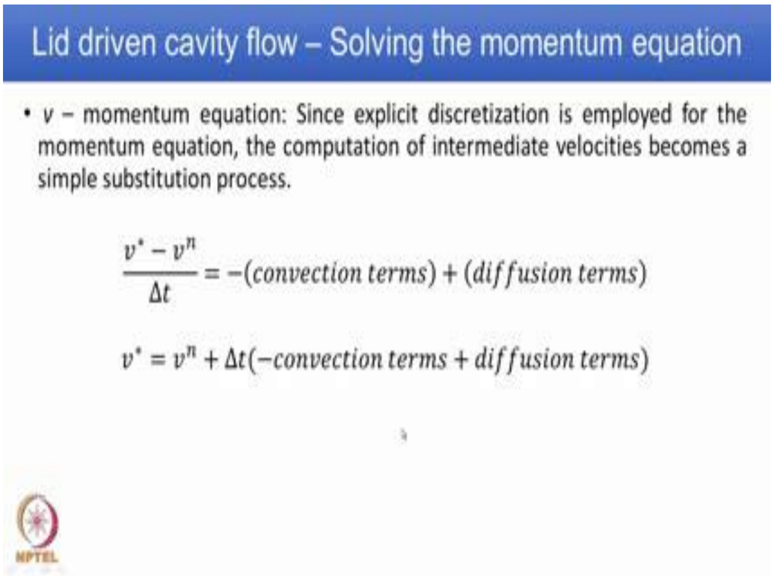
So, in this, we implement central different type of approximation. So, u at east is evaluated from the neighbouring nodes in this way. So, u i comma j plus u i j plus 1 by 2; similarly for the west, the corresponding west is i minus 1, so u i minus 1 j plus 1 u at i minus 1 j divide by 2. Similarly for v, so v at east is evaluated by central differential type of approximation similarly for the other one; and also for north and south. So, once we defined all these, we see corresponding code.

(Refer Slide Time: 16:02)



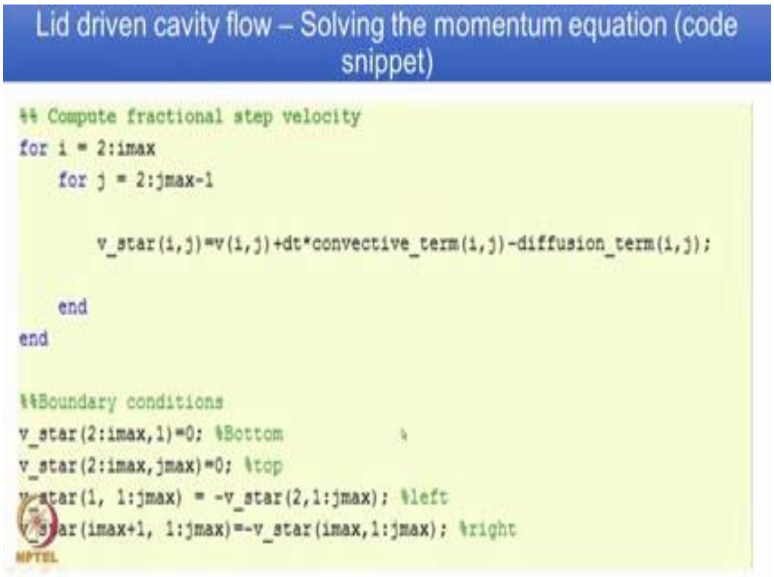
So, for i running from 2 to i max and for j running from 2 to j max minus 1, because on the top that is the last j, we have a boundary condition implemented. So, it is only up to j minus 1; u east, v east then u west, v west, v north, v south all are individually evaluated based on expression we have defined in previous slide. Then all are put together as convective term i comma j equal to minus on for one term plus other term. So, these are individually written here, coefficient for d u v by by dx coefficient for d u v by dx and so on for each line.

(Refer Slide Time: 17:04)



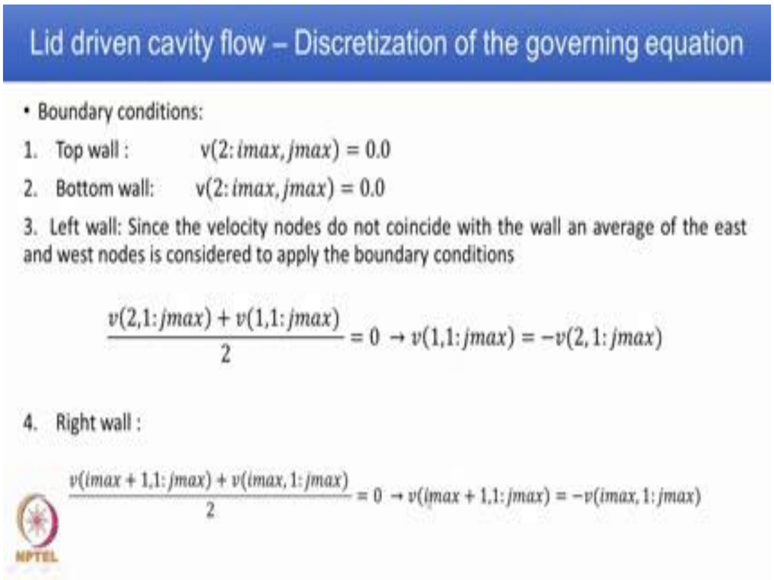
V momentum equation because we have following explicit discretization, the computation of intermediate velocity becomes a straight forward as shown here. So, v star minus v n by delta t equal to minus convection term plus diffusion term. So, we directly used, we have independently evaluated convection term, we have independently evaluated diffusion term. So, we directly used this expression to get what is known as a v star. So, v star is equal to v n plus delta t minus convection term plus diffusion term.

(Refer Slide Time: 17:42)



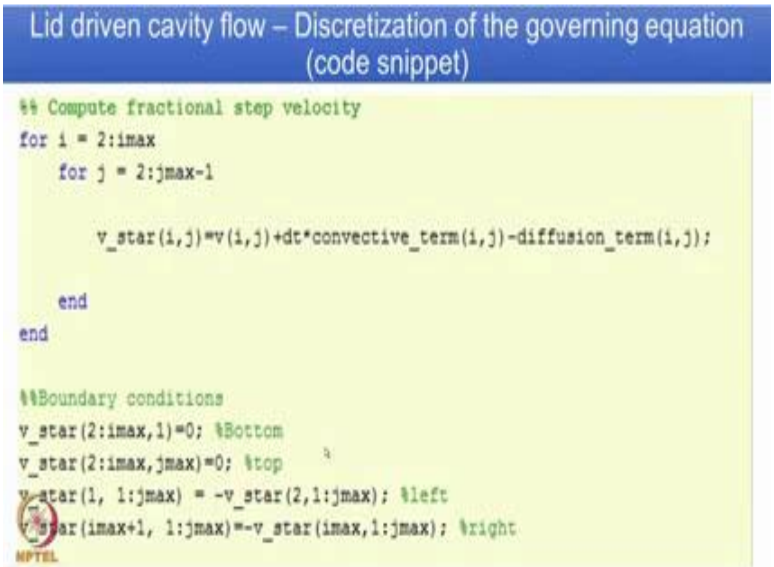
So, compute the fractional step velocity, otherwise v star i for i 2 to i max, and for j 2 to j max minus 1 v star is equal to i comma j plus dt multiply by convection term minus diffusion term and end and end. As we did in u-momentum equation, as soon as we solve particular equation by discretization procedure, we need to enforce boundary condition; we did that for u as u star boundary condition, similarly we should do for v star also.

(Refer Slide Time: 18:23)



So, top wall, v 2 to i max and j max is equal to 0; bottom wall, v 2 to i max comma j max equal to 0; left side wall since the velocity nodes do not coincide with the wall and average of the east and west nodes is considered to apply the boundary condition. So, v 2 1 to j max plus v 1 for i direction and for j direction it is 1 to j max by 2 equal to 0 and if you rearrange this expression as shown here. Similarly on the right side wall v for i it is i max to represent it is the right side wall plus 1 and 1 2 to j max with j direction plus v i max 1 to j max by 2 equal to 0, and if rearrange you get the expression as shown here and this is what we enforce the boundary condition code as shown here.

(Refer Slide Time: 19:32)



So, boundary conditions are v star in i direction 2 to i max and j is 1 equal to zero, this is for bottom wall; v star again i 2 to i max j max equal to 0, this is for the top wall. Please recall on the top wall, we have imposed velocity driving condition, but that is only for u velocity and v velocity is set to zero. Now for the left v star 1 for i and 1 to j max for j equal to minus v star 2 comma 1 to j max this on the left side. Again we should recall what we did for u velocity on the bottom wall, because we do not have a node coinciding with that particular wall, because we are following staggered grid arrangements, we do not have a node coinciding with that wall, we need to have a rearrangements in such a way that boundary condition is actually enforced. So for that only we have a extra ghost cell or ghost node. So, the average between the ghost node and the immediate node adjacent to the boundary condition location that is the left side wall is actually written this form. So, v star 1 comma 1 to j max equal to minus v star 2 1 to j max, this is for node on the right of the left wall and this is from the ghost cell. So, they are equated in such a way, the boundary condition v equal is to zero is enforced on the left side wall. Similarly for the right side wall again using ghost node and ghost cell, we have written line for enforcing boundary conditions for v star. We did this in detail for u star in beginning of this lecture.

So, in this module, we continued the u momentum equation for enforcing boundary condition then we repeated the procedure for v momentum equation. We consider separately convection term, diffusion term, and how to put them together to get v star; once you get solution, we need to enforce boundary condition and we follow the same procedure as we did for u for the right side, left side wall and for the top as well as for the bottom wall. In the next module, we are going to talk about pressure term, projection method and subsequently linking pressure and velocity.

Thank you.