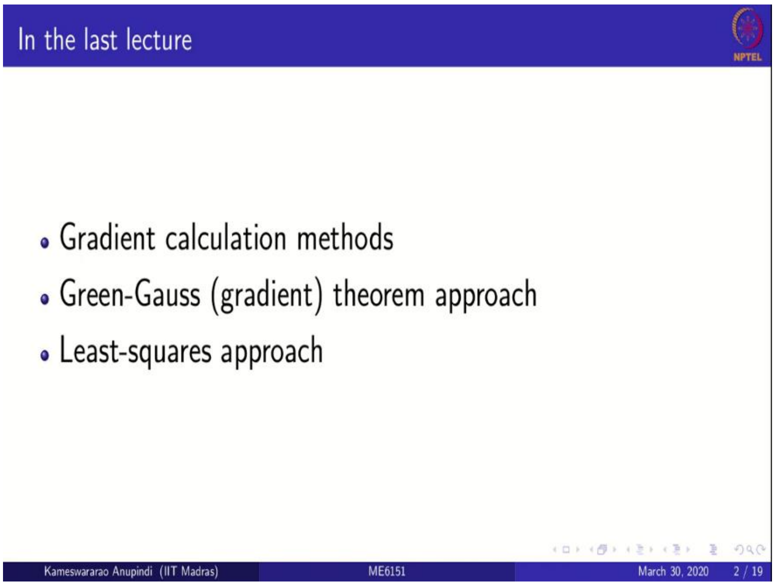
Computational Fluid Dynamics Using Finite Volume Method

Prof. Kameswararao Anupindi
Department of Mechanical Engineering
Indian Institute of Technology, Madras

Lecture – 28
Finite Volume Method for Diffusion Equation: Steady diffusion in unstructured meshes Part 5

Hello everyone, let us get started. So, welcome to another lecture as part of our course ME6151: Computational Heat and Fluid Flow.

(Refer Slide Time: 00:25)



In the last lecture

- Gradient calculation methods
- Green-Gauss (gradient) theorem approach
- Least-squares approach

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 2 / 19

So, in the last lecture or in the previous lectures, we have been looking at gradient calculation methods and we looked at two different methods of gradient calculation; one was the Green-Gauss or the gradient theorem approach and the other one was the least-squares approach.

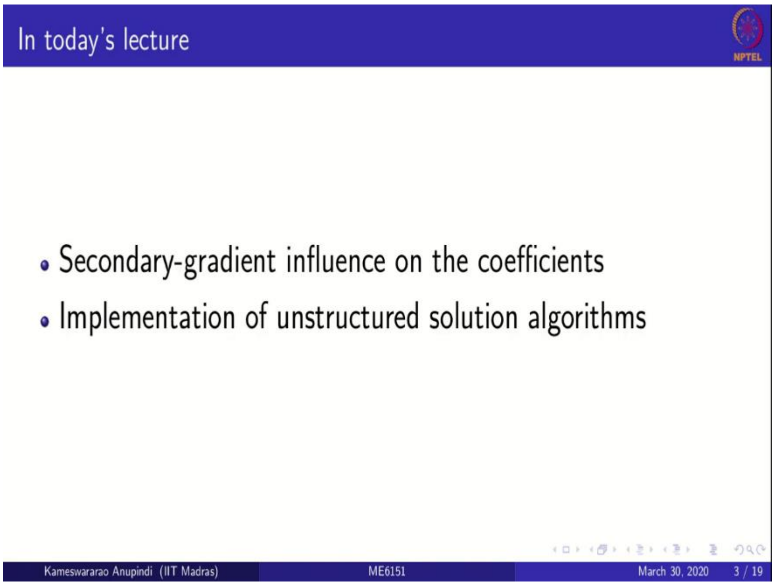
The motivation to look at the Green-Gauss or least-squares approach these gradient calculation methods was that we needed to calculate grad of phi right which was required not only by the secondary gradients that we got as part of the unstructured, non-orthogonal mesh.

But also the gradients of the phi are required from different perspectives right either if you have a looking at a turbulence modeling or if you are working with a non-Newtonian fluid or if you are looking at a large dissimulation all these things would require you to calculate the gradient of the dependent variable grad u or grad phi and so on. So, as a result, we had to calculate gradient at the cell centroids ok.

Then, we looked at these two methods. In the Green-Gauss method, we use the gradient theorem approach or the Green-Gauss theorem wherein the volume integral of the scalar is written as a surface integral right.

And then, we calculated what is the gradient and in the least squares approach, we found a least squares error fit for all the data points and we found one simplified matrix that is a transpose ag equals a transpose b right that was a kind of a 2 by 2 matrix that we have to solve at each and every cell; in order to obtain this vector g which is basically $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial x}$ ok. So, we kind of discussed these things in the last lectures.

(Refer Slide Time: 02:12)



In today's lecture

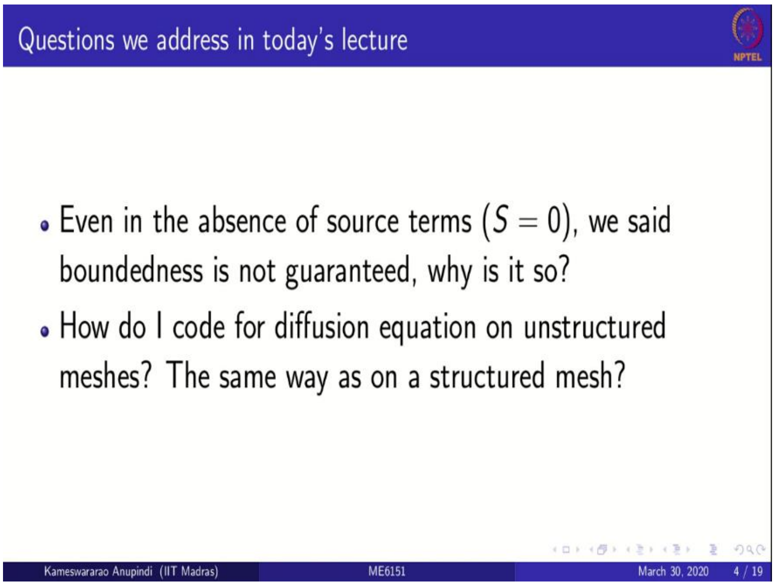
- Secondary-gradient influence on the coefficients
- Implementation of unstructured solution algorithms

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 3 / 19

So, let us move on and so, in today’s lecture, we are going to look at the influence of secondary gradient on the coefficients. So, up till now, we have been saying only that the secondary gradients will make that the solution may not be bounded right that the guarantee that the solution is bounded will be taken away if you have secondary gradients is what we have been saying till now.

But we have not actually proved it or looked into why it may happen so. So, that is one thing we are going to see today. And other thing we are going to look at is how do you implement a solution procedure or how do you program it the solution procedure for solving a diffusion equation on unstructured mesh does not matter whether it is a orthogonal mesh or a non-orthogonal mesh. So, only if you have as long as you have an unstructured mesh, can I go ahead and implement it and what is the way if it is to be done ok. So, this is another thing we are going to look at in today's lecture.

(Refer Slide Time: 03:10)



Questions we address in today's lecture

- Even in the absence of source terms ($S = 0$), we said boundedness is not guaranteed, why is it so?
- How do I code for diffusion equation on unstructured meshes? The same way as on a structured mesh?

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 4 / 19

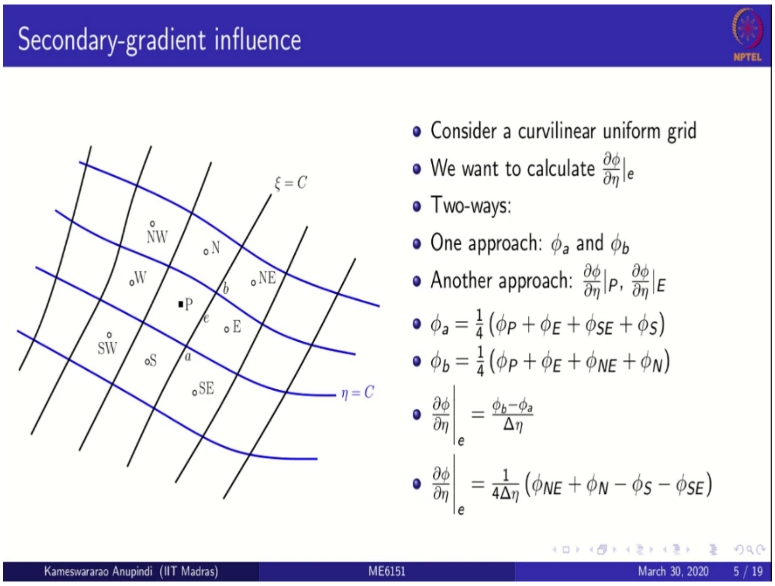
So, let us so, that means, the brand questions we address in today's lecture are the first one is that boundedness is not guaranteed even if the source terms are absent in the context of unstructured non-orthogonal meshes, why is it so? Why is boundedness not guaranteed? That is the first question we will address in today's lecture.

And after finishing that part, we will move on to how do I program for diffusion equation on unstructured meshes ok. Is it the same way I would program on a structured mesh or is there something different? Is there something that can be done differently which will enable us to run the program with let us say; let us say; let us say the program will be faster or it can be; it can be little more memory efficient or it can be better from the algorithm point of view ok.

So, all these things we will kind of consider and then code, how do I code for diffusion equation is the second question we will address in today's lecture ok. So, by the way, this is the this how do I program is how it is done in all the software that that are out there ok.

So, this is the same method that we are going to discuss which is basically belongs to the unstructured solution methods alright.

(Refer Slide Time: 04:23)



Secondary-gradient influence

- Consider a curvilinear uniform grid
- We want to calculate $\frac{\partial \phi}{\partial \eta}|_e$
- Two-ways:
- One approach: ϕ_a and ϕ_b
- Another approach: $\frac{\partial \phi}{\partial \eta}|_P$, $\frac{\partial \phi}{\partial \eta}|_E$
- $\phi_a = \frac{1}{4}\left(\phi_P + \phi_E + \phi_{SE} + \phi_S\right)$
- $\phi_b = \frac{1}{4}\left(\phi_P + \phi_E + \phi_{NE} + \phi_N\right)$
- $\left.\frac{\partial \phi}{\partial \eta}\right|_e = \frac{\phi_b - \phi_a}{\Delta \eta}$
- $\left.\frac{\partial \phi}{\partial \eta}\right|_e = \frac{1}{4\Delta \eta}\left(\phi_{NE} + \phi_N - \phi_S - \phi_{SE}\right)$

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 5 / 19

Let us move on with the first question. Let us look at the influence of secondary gradients on the coefficients ok. So, here we consider a on the left-hand side, we show a mesh here. This is basically what I would like to call it as a curvilinear mesh, but that is uniform so; that means, these cells are pretty much uniform, they are of the same size approximately, but then, the mesh need not be orthogonal ok, the mesh is non-orthogonal.

For example, if you connect the cell centroids, the area may not be in the exact same direction. So, we are choosing this simplified geometry in order to understand the effect of influence. So, we are looking at a uniform mesh that is non or need not be orthogonal that is non-orthogonal.

So, as usual we have our cell P that is right here and it also has the east cell, west cell, north cell and the south cell and we also have the corner neighbour’s here which are the northeast, northwest, southeast and southwest and also we have the east face which has two vertices a and b, this is the same nomenclature we have been following till now.

And we also have these blue lines which are basically η equal to constant and the black lines which are basically ξ equal to constant; that means, if you move along the ξ equal to constant, η increases and if you move along this η equal to constant, ξ increases. So, the secondary gradients would lead to the calculation of $\frac{\partial \phi}{\partial \eta}$ on the east face right.

Now, how do we calculate this? We looked at this calculation method a while back uh, we can do it in two ways; one is basically calculate what is ϕ_a and ϕ_b, essentially interpolate ϕ_a from its neighbor's that is a P east, south east and south and interpolate ϕ_b from P east, north east and north and then using these interpolate values at a and b, we can calculate what is $\frac{\partial \phi}{\partial \eta}$ in this direction on this faced e; on this face e right.

Now, another approach is of course, to directly calculate the gradients at the adjacent cell centroids that is $\left.\frac{\partial \phi}{\partial \eta}\right|_P$ at P and $\left.\frac{\partial \phi}{\partial \eta}\right|_E$ at E right calculate it at this location and at this location and then, linearly interpolate them. So, we take the first approach here in order to understand.

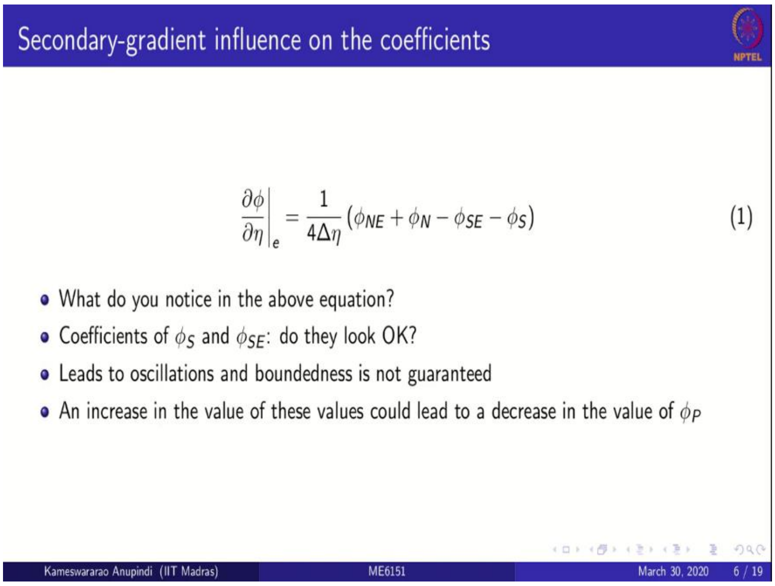
So, in the first approach, we can write ϕ_a as ϕ_P plus ϕ_E plus ϕ_{SE} plus ϕ_S divided by 4 because we are saying this is a uniform mesh. If it were; if it were not uniform mesh, then you would get instead of one-fourth, you would get these lengths right that are multiplying these length ratios that are multiplying ϕ_P, ϕ_E, ϕ_{SE} and so on ok.

Similarly, ϕ_b can be calculated as one-fourth of the neighbor's that is ϕ_P, ϕ_E, ϕ_{NE} and ϕ_N ok. So, essentially, we got values for phi and ϕ_b from linear interpolation. Now, what is the definition for $\frac{\partial \phi}{\partial \eta}$ on the east face? That is nothing, but ϕ_b minus ϕ_a upon Δη right. So, that is $\left.\frac{\partial \phi}{\partial \eta}\right|_e$ equals ϕ_b minus ϕ_a by Δη.

Now, we plug in what is ϕ_b and ϕ_a from the values we have obtained before; that means, we plug it in these two, then this would give us to ϕ_P gets cancelled when you say ϕ_P minus ϕ_a here ϕ_P gets cancelled, similarly ϕ_E gets cancelled and we are left with these two quantities with a positive sign and these two with a negative sign right divided by 4 times Δη.

So, what we get for $\frac{\partial \phi}{\partial \eta}$ on the east face is 1 upon 4 Δη times ϕ_{NE} plus ϕ_N minus ϕ_S minus ϕ_{SE} that is what we get alright fine.

(Refer Slide Time: 08:39)



Secondary-gradient influence on the coefficients

$$\left.\frac{\partial \phi}{\partial \eta}\right|_e = \frac{1}{4\Delta\eta}\left(\phi_{NE} + \phi_N - \phi_{SE} - \phi_S\right) \qquad (1)$$

- What do you notice in the above equation?
- Coefficients of ϕ_S and ϕ_{SE}: do they look OK?
- Leads to oscillations and boundedness is not guaranteed
- An increase in the value of these values could lead to a decrease in the value of ϕ_P

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 6 / 19

So, this is what we got for the secondary gradient that is $\frac{\partial \phi}{\partial \eta}$ on the east face which is 1 4 1 over 4 $\Delta\eta$ of $\phi_{NE}\, plus$ ϕ_N minus ϕ_{SE} minus ϕ_S ok. Now, what do you notice in this equation? Is does this equation look fine? Well, it is fine from mathematical approach point of view, but is it fine from our finite volume method point of view?

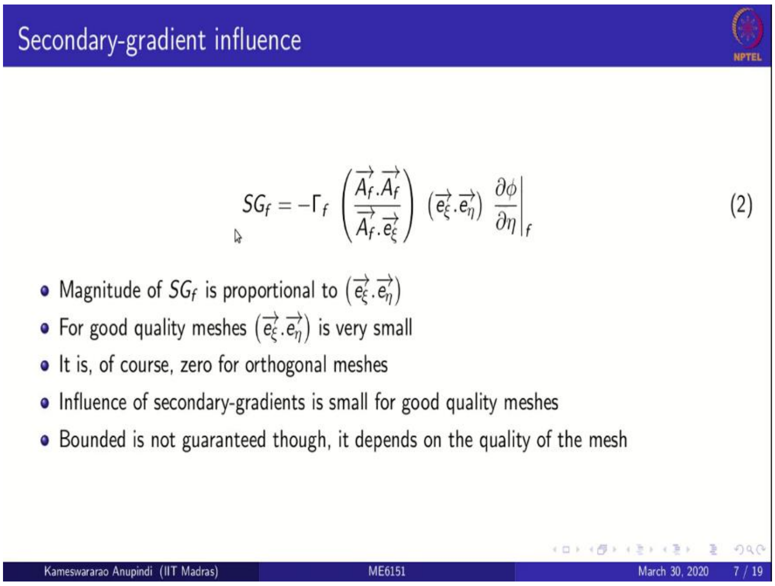
What about the coefficients for ϕ_S and ϕ_{SE}? Do they look ok? Do they look fine? No, they do not look fine because there are there is a negative coefficient that is multiplying them that is the coefficient for ϕ_{SE} would be 1 by 4 $\Delta\eta$ right.

So, similarly for ϕ_S would be 1 by 4 $\Delta\eta$. So, this negative coefficients is not good for us because if we have negative coefficients, then what it means is that if this value goes up; goes up in the previous iteration, then the value of ϕ_b right which is basically calculated because of $\frac{\partial \phi}{\partial \eta}$ can go down because of there is a negative sign right.

So, essentially this kind of behavior is not encouraged in our finite volume especially when you have, when you are solving diffusion equation alone right. So, essentially this will lead to oscillations and this is the reason why boundedness is not guaranteed right because an increase in the value of ϕ_{SE} or ϕ_S could lead to a value decrease in value of ϕ_P at a next iteration right. So, this leads to oscillations and as a result the boundedness is property is not guaranteed.

If you go back, all the coefficients we had before were all positive right. If you look at $a_P \phi_P = \sum a_{\mathrm{nb}} \phi_{nb} + b$ all the a_{nb}'s were positive right, but that is not the case here which will lead to oscillations. So, that is the reason why we said boundedness is not guaranteed if you have secondary gradients; that means, if you have a non-orthogonal mesh, boundedness is not guaranteed ok, but then that is not so bad.

(Refer Slide Time: 11:06)



Secondary-gradient influence

$$SG_f = -\Gamma_f \left(\frac{\vec{A_f}.\vec{A_f}}{\vec{A_f}.\vec{e_\xi}}\right)(\vec{e_\xi}.\vec{e_\eta})\,\frac{\partial \phi}{\partial \eta}\bigg|_f \qquad (2)$$

- Magnitude of SG_f is proportional to $(\vec{e_\xi}.\vec{e_\eta})$
- For good quality meshes $(\vec{e_\xi}.\vec{e_\eta})$ is very small
- It is, of course, zero for orthogonal meshes
- Influence of secondary-gradients is small for good quality meshes
- Bounded is not guaranteed though, it depends on the quality of the mesh

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 7 / 19

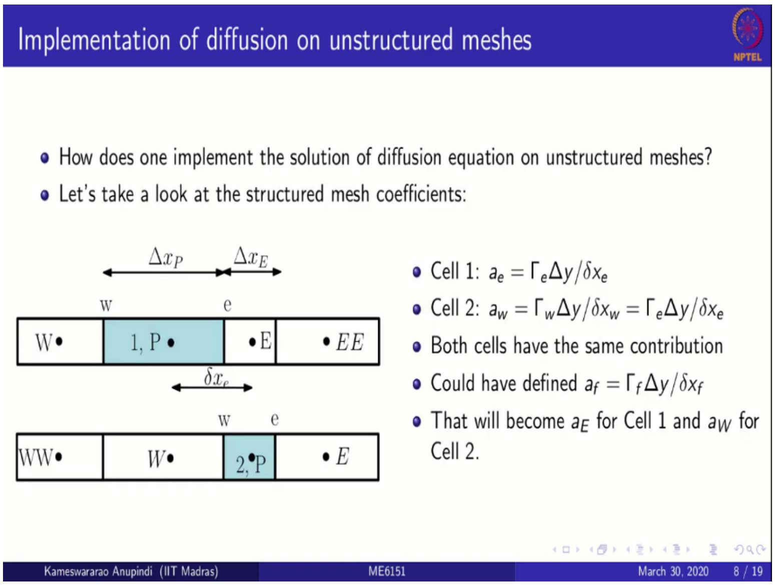
So, let us see what is the total value of the secondary gradient, it is not just $\frac{\partial \phi}{\partial \eta}$ right. So, essentially if we go right what is the secondary gradient value that is SG f equals minus Γ_f A_f dot A_f by A_f dot e_ξ; e_ξ dot e_η $\frac{\partial \phi}{\partial \eta}$ on the face east right, this is the total value of the secondary gradient that means, the magnitude of secondary gradient is proportional to so, whatever you calculate here times this guy times is proportional to e_ξ dot e_ξ right, this is the controlling factor which will contribute what percentage of the value of the secondary the $\frac{\partial \phi}{\partial \eta}$ goes into the second gradient right.

So, for example, if you have an orthogonal mesh, this is of course, 0; that means, whatever you calculate does not go into the property at all, but if you have a non-orthogonal mesh, then if you have a good quality mesh, then this will still be very close to 0 let us say it is 0.1 or 0.05 or something close to 0. As a result, the secondary gradients influence will be very small on the solution for good quality meshes ok. So, that is one reason why you should have a mesh that is as much orthogonal as possible in almost all the cells ok. So, that is one reason to avoid boundedness problems.

Therefore, but anyway even if it is small, effect is small, the boundedness is not guaranteed because it depends on the quality of the mesh. So, it is not absolute. For example, in the context of orthogonal meshes, we said boundedness is always guaranteed right irrespective of the mesh I mean of course, the mesh is always orthogonal ok.

So, but in the context of non-orthogonal meshes, boundedness is not guaranteed, but if you have a good quality mesh, the influence of second gradients is small as a result, you would not get very large oscillations in the solution ok. So, that is the influence of secondary gradient alright. So, that answers our first question, why boundedness is not guaranteed if you have a non-orthogonal mesh.

(Refer Slide Time: 13:20)



Implementation of diffusion on unstructured meshes

- How does one implement the solution of diffusion equation on unstructured meshes?
- Let's take a look at the structured mesh coefficients:

- Cell 1: $a_e = \Gamma_e \Delta y/\delta x_e$
- Cell 2: $a_w = \Gamma_w \Delta y/\delta x_w = \Gamma_e \Delta y/\delta x_e$
- Both cells have the same contribution
- Could have defined $a_f = \Gamma_f \Delta y/\delta x_f$
- That will become a_E for Cell 1 and a_W for Cell 2.

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 8 / 19

Let us move on to the second question that we wish to answer in this lecture today that is how do I program for diffusion equation on unstructured meshes? Well, of course, one way is to go about the way you are doing it right now I mean for structured measures that is basically again start off with the equation $a_P \phi_P = \sum \mathrm{a}_{\mathrm{nb}} \phi_{nb} + b$ and write such an equation for each and every cell right.

So, for each and every cell, you would write such an equation and then, you will go cell by cell. You will start from let us say cell 1 that you have and then, move on to cell 2, cell

3 and so on and when you go to every cell, you would calculate what is a and b. Let us say cell 1 has four neighbors.

You will calculate what is a 1, a 2, a 3, a 4 all these four neighbor's and then, you would calculate what is a_P, you calculate what is b, the contributions coming from the source terms and then, you assemble this equation right numerically and then, once you have all the equations, you can use some method Gauss-Seidel or something and solve for it.

But what we see is that this is not a very good way, or it is not an ideal way if you have to code up for an unstructured mesh ok. Why is it so? One is from the calculation perspective; the other is from the solution perspective ok. So, essentially from the point of view of the algorithm or what we have devised, it is better to do it in a different way rather than visiting each and every cell and also from the point of view of calculations, the number of calculations involved can also be reduced if you do it in a; in a different way ok.

Now, essentially coming back to this question, how do I implement solution of diffusion equation unstructured measures, now in order to understand this, let us first take a look at a structured mesh ok. So, I am basically looking at a structured mesh that is one-dimensional ok, but that is non-uniform, you can see the widths are all different for every cell.

Now, what we see here on the top is the same mesh as what we see at the bottom ok. So, these are basically the same mesh. Only thing is that in the top row here, we are solving for the P cell that is this bigger cell 1 and in the bottom row here we are solving for this neighbor of this guy that is 2. This is our primary cell ok.

So, that is what we are solving for and the deltas for these cells are Δx_P and Δx_E, these are the widths of the cells and similarly when you are solving for P face, we have east and west and the neighbors are east and west for the P cell and of course, this cell is east to the east. So, we call it east-east and we are here for the second cell, this is basically the primary cell, then this is our east cell, this is our west cell and this is our west of west ok. So, we just change the nomenclature here, but it is the same mesh we are talking about in both of these things and then distance between the P cell and the E cell is Δx_E ok.

So, let us see if you are writing, if you are discretizing the cell 1, what is our a_E; what is our a_E? a_E is nothing, but $\frac{\Gamma_e \Delta y}{\delta x_e}$ right. So, that is Γ_e on the face and Δy this height and δx_e

is this distance ok. Now, when you; when you are done with cell 1, when you go to cell 2, what would be the coefficient for a west? a_W would be Γ_w that is on this particular face Δy this height times δx_w that is this width.

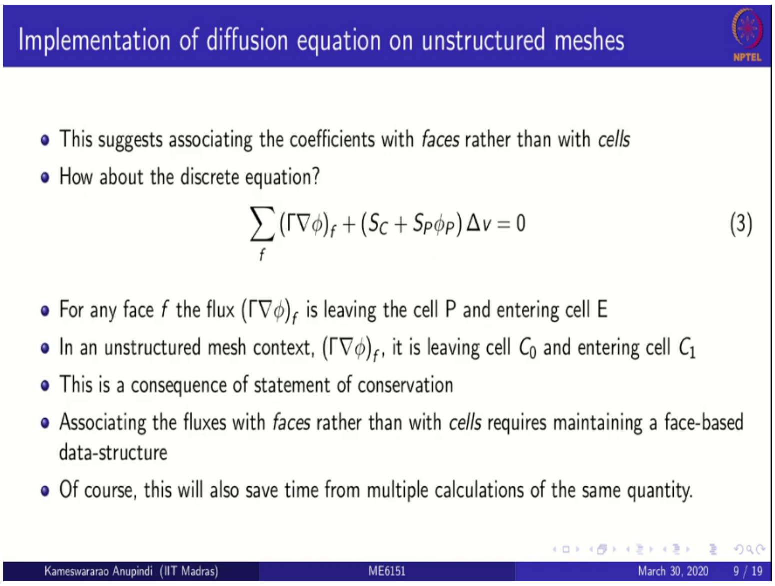
Now, what we see is that this is we are just calling with a different name, but the numerical values of these are same as what we have before because we are referring to the same face right. So, essentially, a_W is nothing, but same as $\frac{\Gamma_e \Delta y}{\delta x_e}$ right. It is basically the same face, same distance, same gamma we are referring to, but it is now going into the coefficient for west coefficient for the next cell right so; that means, we are actually duplicating these calculations because we are doing using the same numbers, calculating the same fraction and then doing it twice.

Once when we are at cell 1 and we are calling it an a east and second one when we are at cell 2 and we are calling it a west so, we do not have to do this thing. If I can do it once for the face and kind of send this contribution to the neighboring cells, then I can actually save one multiplication and one division here right which are extra which need not be done that will certainly speed up the calculation.

And also, there is another way to look at this thing. This is basically what we are saying is that this these coefficients have come out to be the same because of some under ruling principle. What is the principle? Basically, the conservation of fluxes right whatever is the flux that is leaving here is entering this cell that is why these coefficients have come out to be same right because your gamma grad phi on the face is basically leaving from cell 1 and we it is entering cell 2, that is why it is the same flux that left cell 1 entering cell two and as a result, the coefficients are the same ok.

So, what we just looked at is basically both the cells have the same contribution ah. So, instead of calling them with east and west, we could have defined a_f which is basically calculate $\frac{\Gamma_e \Delta y}{\delta x_e}$ and the same a f can be given contributed to as a east for cell 1 and a west for cell 2 right we could have done that, that is essentially what we plan to do or what we do for unstructured meshes ok. So, once I think this part is clear, the remaining things are straight forward ok.

(Refer Slide Time: 19:24)



Implementation of diffusion equation on unstructured meshes

- This suggests associating the coefficients with *faces* rather than with *cells*
- How about the discrete equation?

$$\sum_f (\Gamma \nabla \phi)_f + (S_C + S_P \phi_P)\,\Delta v = 0 \qquad (3)$$

- For any face f the flux $(\Gamma \nabla \phi)_f$ is leaving the cell P and entering cell E
- In an unstructured mesh context, $(\Gamma \nabla \phi)_f$, it is leaving cell C_0 and entering cell C_1
- This is a consequence of statement of conservation
- Associating the fluxes with *faces* rather than with *cells* requires maintaining a face-based data-structure
- Of course, this will also save time from multiple calculations of the same quantity.

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 9 / 19

So, let us move on. So, what this kind of discussion suggests is that we can associate the coefficients with the faces rather than with the cells right. Instead of calling a_E anda_W to belong to the cells, we can say that these actually belong to the faces right, the common face defines what will be the coefficient for east and what will be the coefficient for west.

Now, let us look at the discrete equation. So, in the derivations, we got this discrete equation which is sigma f, $(\Gamma\nabla\phi)_f$ plus $(S_C + S_P\phi_P)\Delta V$ equal to 0 right. So, this is the flux right that we got. So, for any face, the flux $\Gamma\nabla\phi$ is leaving the cell P and entering the cell east right that is what we have, it is basically conservation of fluxes times the source term is balanced right that is the conservation statement.

Now, instead of if you have instead of having a structured mesh, if you have an unstructured mesh, then $\Gamma\nabla\phi$ for a particular face is leaving the cell C_0 and it is entering the cell C_1 right or it is leaving the cell C_0 and it is entering the cell C_2 right some other cell and so on ok. Now, this is because of the statement of conservation right. Because the same flux is whatever is leaving cell 0 is entering cell C_1.

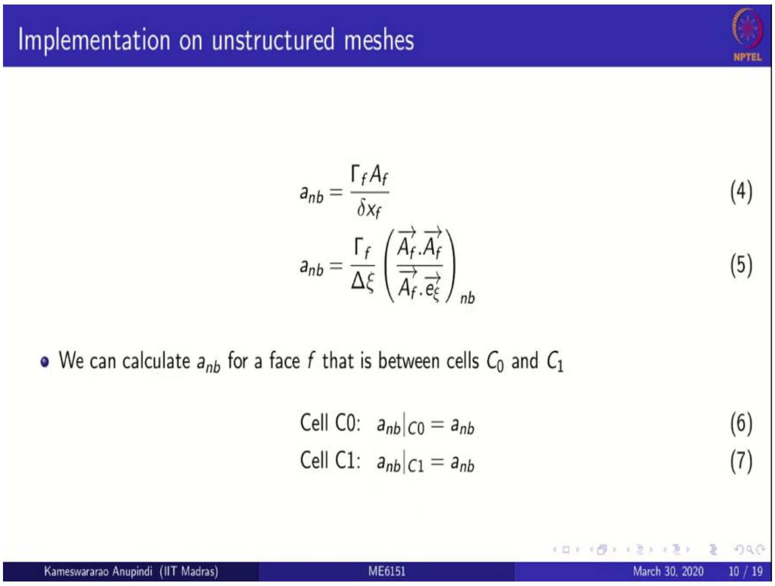
Now, associating the fluxes with faces right rather than with the cells requires of course, this requires you to maintain a list of your faces; that means, you need to have a face-based data-structure. Let us say up till now, you only had a cell based data structure. For example, you had I have 10 cells so, I will call my a east with a east of 10 in which your 0, 1, 2 all these things led to different cells.

Now, you would call them as faces as well right, you need to maintain a face based data structure; that means, you need to know what is the faces you have and which faces have what cell neighbors right for example, face f has some cell neighbor C_0 and cell C_1, next face f 2 has cell C_0 and cell C_2 and so on right. So, that connected information you need to know when you are dealing with unstructured measures ok. Let us say we know that from some mesh generation software or whatever you write ok.

Now, of course, this method of associating the coefficients to the faces rather than to the cells saves us from multiple calculations of the same quantity right because this gamma grad phi that we calculate for one face is basically going into two different cells ok. So, that way we can say computation that let us say per cell we can you save maybe in this context we can save like two, one multiplication and one division essentially, two floating point operations.

Of course, it is it depends on if it is unstructured you have more terms here to evaluate as well and also it is prone to lesser errors because now you have the same flux that you calculate is going into both of them. So, it is kind of probably less error alright.

(Refer Slide Time: 22:42)



Implementation on unstructured meshes

$$a_{nb} = \frac{\Gamma_f A_f}{\delta x_f} \qquad (4)$$

$$a_{nb} = \frac{\Gamma_f}{\Delta\xi}\left(\frac{\vec{A_f}.\vec{A_f}}{\vec{A_f}.\vec{e_\xi}}\right)_{nb} \qquad (5)$$

- We can calculate a_{nb} for a face f that is between cells C_0 and C_1

$$\text{Cell C0:} \quad a_{nb}|_{C0} = a_{nb} \qquad (6)$$

$$\text{Cell C1:} \quad a_{nb}|_{C1} = a_{nb} \qquad (7)$$

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 10 / 19

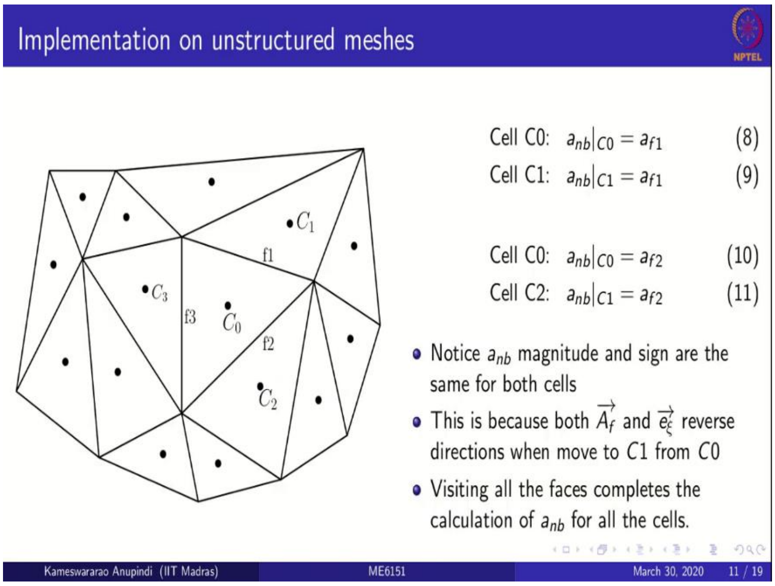
So, with this motivation now, let us look at what how do you calculate the a neighbor ok. So, a neighbor is basically, if you have a structured mesh that is as shown in equation 4, a_{nb} equals $\frac{\Gamma_f A_f}{\delta x_f}$ that we have already seen. But, if you have an unstructured mesh, what will be your $\mathrm{a_{nb}}$? $\mathrm{a_{nb}}$ is basically coming from your primary gradient coefficient right, from the primary gradient term that is $\mathrm{a_{nb}}$ equals $\frac{\Gamma_f}{\Delta\xi}\frac{A_f\cdot A_f}{A_f\cdot e_\xi}$ right that is your primary gradient coefficient that is $\mathrm{a_{nb}}$ ok.

Now, we can of course, calculate $\mathrm{a_{nb}}$ for a face f that is between cell C_0 and C_1 right just like we have calculated for cell P and cell east, we can calculate this for a particular face because when I am in a particular face, I know what is gamma on the face, I know what is $\Delta\xi$ the distance between C_0 and C_1, I know what is $\overrightarrow{\mathrm{A_f}}$ and e_ξ.

Now, once I know all of these things, I can calculate what is $\mathrm{a_{nb}}$. Now, this $\mathrm{a_{nb}}$ contribution has to be given to both the cells that share the face right. So, these cells C_0 and C_1 share the face. So, this $\mathrm{a_{nb}}$ I calculated up in equation 5 here, I would simply assign it to the $\mathrm{a_{nb}}$ of C_0 cell and also I will assign it to $\mathrm{a_{nb}}$ of C_1 cell right. So, essentially, these two are different, this is $\mathrm{a_{nb}}$ for C_0 cell and this is $\mathrm{a_{nb}}$ for C_1 cell ok.

Now, see we are we are not going cell by cell, we are going by face and we are accessing cells from the faces right that is what we are doing ok that is so far so good. So, we found a way how to do this thing.

(Refer Slide Time: 24:38)



Implementation on unstructured meshes

Cell C0: $a_{nb}|_{C0} = a_{f1}$ (8)

Cell C1: $a_{nb}|_{C1} = a_{f1}$ (9)

Cell C0: $a_{nb}|_{C0} = a_{f2}$ (10)

Cell C2: $a_{nb}|_{C1} = a_{f2}$ (11)

- Notice a_{nb} magnitude and sign are the same for both cells
- This is because both $\overrightarrow{A_f}$ and $\overrightarrow{e_\xi}$ reverse directions when move to $C1$ from $C0$
- Visiting all the faces completes the calculation of a_{nb} for all the cells.

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 11 / 19

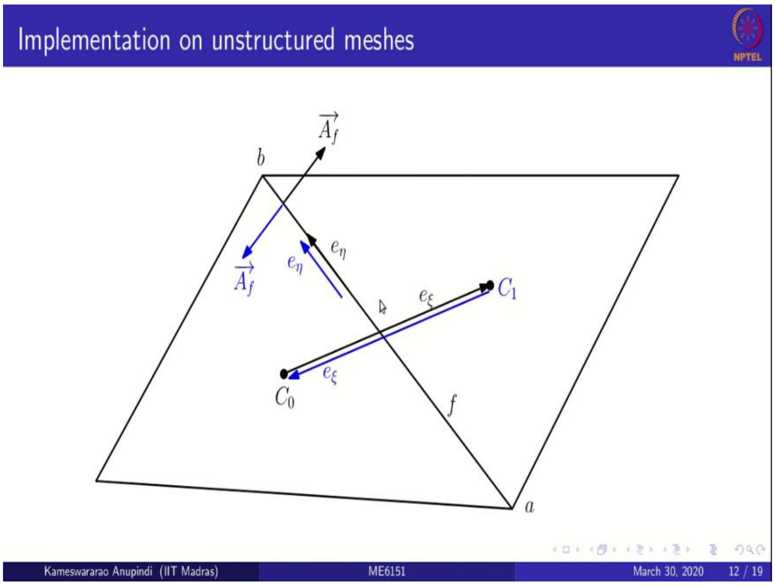
Now, let us say we have an unstructured mesh that is shown on the left hand side here something like this where C_0 is the primary cell we are focusing on and then, we have these three faces that is f1, f2, f3 and which share cells C_1, C_2, C_3 ok. So, that means, when I calculate a f1, I need to add, assign that value to a_{nb} for C_0 and a_{nb} for C_1, a_{nb} we mean a_{nb} for C_0 to C_1 and a_{nb} for C_1 we mean C_1 to C 0 right.

For example, this a_{nb} can be a 01 right that is basically coefficient of the 0 for one cell right. Similarly, this can be a 10 this could be coefficient of one, 1st cell to the 0th cell right. So, that is a_{nb} for C_1. We will assign this for both the cells. Similarly, when we go to the next face f 2, we calculate what is a f2 from the original expression that is $\frac{\Gamma_f}{\Delta\xi}\frac{A_f \cdot A_f}{A_f \cdot e_\xi}$ and then, assign it to a_{nb} for C_0 and a_{nb} for this is; this is for a_{nb} for C_2, this also should read C_2 not C_1 alright.

Now, what we see here is that the magnitude and sign of a_{nb} is the same for both the cells right. f1 we are not changing anything; we are assigning the same value to both C_0 and C_1. Similarly, whatever you calculate for a f2, we are assigning it to C_0 and C_2 right we are not changing why is this?

Why is this? Actually, this is happening because although we calculate from a_{nb} from this cell, when we go here there are some things certain things that are changed right for example, you're a f now points this way and so on. So, this contribution is the same for both cells because if we look at let us say two triangles in isolation that is on the next page here ok.

(Refer Slide Time: 26:49)



Implementation on unstructured meshes

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 12 / 19

So, here we have C_0 cell, C_1 cell and when you have and the black lines here indicate the black arrows indicate the entire nomenclature $\overrightarrow{\mathrm{A_f}}$, e_ξ, e_η with respect to C_0 cell that is also shown in black and blue lines indicate $\overrightarrow{\mathrm{A_f}}$, e_η, e_ξ with respect to C_1 cell. So, how would you draw this when you are on, when you consider C_1 as your primary cell and how will you draw this when you consider C_0 as your primary cell ok. So, that is what is black and blue lines indicate.

Now, we are calculating $\mathrm{a_{nb}}$ from that is basically we are calculating A_f when we are sitting on C_0 right with respect to C_0 because let us assume that we are calculating with respect to C_0; that means, our e_ξ points in this direction, the black direction and A_f points in this direction and then we calculate some a nb. Now, that $\mathrm{a_{nb}}$ we are saying is the same as the $\mathrm{a_{nb}}$ you would calculate, when you are sitting on C_1 and you calculate from f.

Because when you calculate from C_1 perspective from this f, A_f actually points this way and your e_ξ now points this way right, but this would come out to be the same because if you go back to this calculation, what is the primary gradient contain $\mathrm{a_{nb}}$ contain? Γ_f, this is the same for both the cells whether you sit on C_0 and C_1, $\Delta\xi$ is the same, what about $\overrightarrow{\mathrm{A_f}}$ dot $\overrightarrow{\mathrm{A_f}}$? $\overrightarrow{\mathrm{A_f}}$ dot $\overrightarrow{\mathrm{A_f}}$ although it is changing direction right from black to blue, but this magnitude is always the same because the squared quantity.

Now, what about $\overrightarrow{\mathrm{A_f}}$ dot e_ξ? $\overrightarrow{\mathrm{A_f}}$ dot e xi is something that we have to look at. $\overrightarrow{\mathrm{A_f}}$dot e_ξ, when you calculate from C_0 is basically pointing this way and e_ξ pointing this way right. So, there is some acute angle between them now when you switch to C_1 what would this be? This would be $\overrightarrow{\mathrm{A_f}}$ is pointing this way and e_ξ is pointing this way. So, both of them flip their sign as a result and they maintain the same angle right.

As a result, $\overrightarrow{\mathrm{A_f}}$ the dot product of $\overrightarrow{\mathrm{A_f}}$ and e_ξ would come out to be the same and as a result, whatever we calculate on the face with respect to either cell would be the same contribution for both the cells right that is basically $\mathrm{a_{nb}}$ would be the same whether you are sitting on C_0 and calculating it on C_1 and calculating it.

Because $\overrightarrow{\mathrm{A_f}}$ dot e_ξ both the vectors change sign and as a result, this will come out to be the same that is something you have to verify once again alright. So, because both $\overrightarrow{\mathrm{A_f}}$ and e_ξ

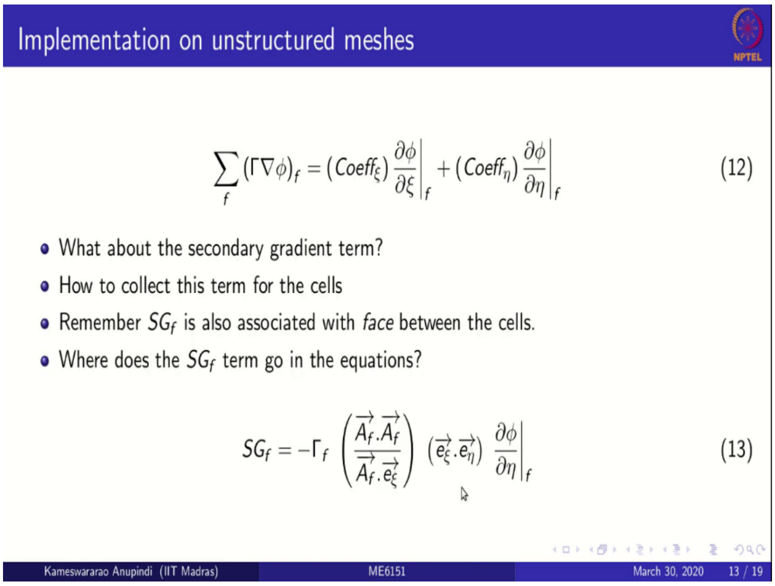
reverse directions, when we move from C_1 to C_0 this quantity a nb would remain the same whether you calculate from C_0 side or C_1 side for the face f.

Now, what now what we need to do? We need to of course, visit each and every face in this fashion. Once you visit each and every face in this fashion, when you finish visiting all the faces that completes the calculation of $\mathrm{a_{nb}}$ for all the cells. For example, this cell has one a_P that is a 0 and it has three neighbors so, they will have a 01, a 02, a 03 right three neighbors, but three neighbors will be populated when you have already visited f1, f2 and f3 right. Similarly, once you; once you visit all the faces, then for every cell their neighbor coefficients would be have been filled already right by the time you finish, you have filled all the neighbor coefficients for all the cells right.

Do you see the difference? If in the original cell centered or cell-based approach you would go to every cell and populate the neighboring coefficients for every cell. But in this case, you are visiting faces and populating in pairs the neighbor coefficients for those cells which are shared, which are sharing that face right. So, the pattern is different, but once you finish all the visiting all the faces, you have already assigned all these cell neighboring coefficients ok. This is faster, do you see why?

This is faster because you are not calculating again for every cell and also you are assigning both of them at once right rather than going here and calculating three and coming here and calculating three, you are going here and assigning two of them right. So, by the time you finish three faces, you actually assigned 6 coefficients right fine alright. Then, let us move on.

(Refer Slide Time: 31:40)



Implementation on unstructured meshes

$$\sum_f (\Gamma \nabla \phi)_f = (Coeff_\xi)\frac{\partial \phi}{\partial \xi}\bigg|_f + (Coeff_\eta)\frac{\partial \phi}{\partial \eta}\bigg|_f \tag{12}$$

- What about the secondary gradient term?
- How to collect this term for the cells
- Remember SG_f is also associated with *face* between the cells.
- Where does the SG_f term go in the equations?

$$SG_f = -\Gamma_f \left(\frac{\vec{A_f}.\vec{A_f}}{\vec{A_f}.\vec{e_\xi}}\right)(\vec{e_\xi}.\vec{e_\eta})\frac{\partial \phi}{\partial \eta}\bigg|_f \tag{13}$$

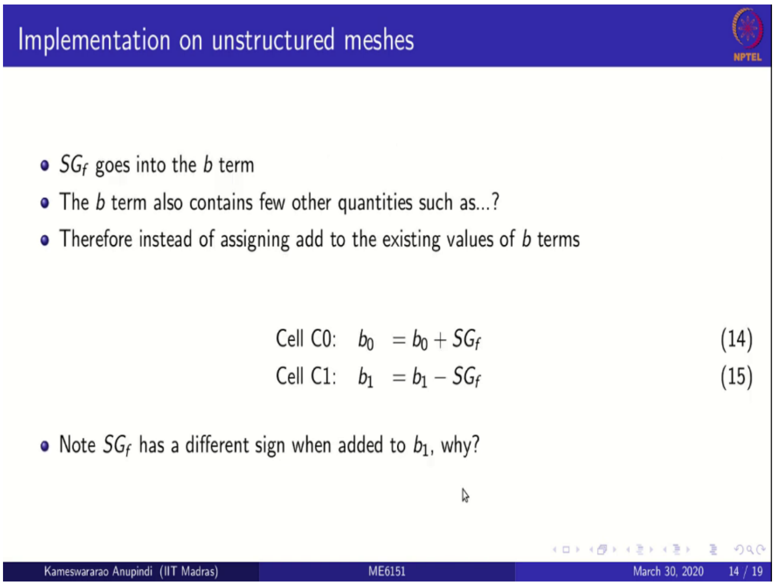
Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 13 / 19

So, this is about the primary gradient right, we are talking about in a non-orthogonal mesh then, you also have the secondary gradient coming into play right because we said implementation on unstructured meshes, you have gamma grad phi can be written as some coefficient times $\frac{\partial\phi}{\partial\xi}$ plus some coefficient times $\frac{\partial\phi}{\partial\eta}$ right. So, you have these two. So, we have already looked at how to deal with $\frac{\partial\phi}{\partial\xi}$.

Now, what about the secondary gradient? Now, how do you calculate this term for the cells? Of course, you can go to again every cell and calculate, but that is not the motivation, we want to calculate from the face-based data-structure because SG is also associated with the face between the cells because that is why we write secondary gradient sub f right. So, this is also associated with the face. So, it would save us computational time if you can calculate it once and assign it to both the cells that share the face ok.

So, where does the; where does the secondary gradient term go in the equations? So, we know the second integration is minus Γ_f A_f dot A_f by A_f dot e_ξ dot e_η $\frac{\partial\phi}{\partial\eta}\Big|_f$.

(Refer Slide Time: 33:00)



Implementation on unstructured meshes

- SG_f goes into the b term
- The b term also contains few other quantities such as...?
- Therefore instead of assigning add to the existing values of b terms

$$\text{Cell C0:} \quad b_0 = b_0 + SG_f \tag{14}$$

$$\text{Cell C1:} \quad b_1 = b_1 - SG_f \tag{15}$$

- Note SG_f has a different sign when added to b_1, why?

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 14 / 19

Now, why does the secondary gradient term go? Does it go into the matrix a? No, it does not. It goes into the right-hand side right that is the b term; it goes into the b term fine ok. So, SG f goes to the b term; that means, the b term also contains not only the secondary gradient term, but also other quantities.

What are the other quantities that the b terms for the cells contain? They contain the source term as c times delta v of course, and also, they also contain some contribution from the any Dirichlet boundary conditions if you have right. So, those also will go into the b term. So, as a result, we cannot simply assign the secondary gradient into the b term of any cell rather we have to add to the existing b value ok.

Therefore, instead of assigning, add to the existing value because the b term already contain some values from the boundary conditions, from the source terms and so on so, you cannot overwrite it with only the secondary gradient term ok. So, as a result, I would calculate what is the secondary gradient term for a particular face and I would add that contribution to the C_0 cell, I will add b equals b the existing value of let us say source terms Dirichlet boundary conditions and so on plus second gradient. For cell 1, I would subtract the second gradient from whatever I have calculated for the particular face while sitting on cell 0.

Why do I subtract it here? Why not add it just like the primary gradient? Ok. So, this is again goes back to the figure ok, we have this figure where the black lines are all definitions for $\overrightarrow{\mathrm{A_f}}$, $\overrightarrow{\mathrm{A_f}}$, e_η for C_0 cell and the blue lines are the definitions when you are sitting on the C_1 cell. Now, what is the, we are only calculating let us say from one cell and then, we are trying to assign it for both the cells right that is the issue here.

Now, one thing we have to keep in mind is that we are keeping the a e_η direction the same for both the cells ok. I am assuming that e_η is the same direction as we have had before this is one assumption we are doing. Now, what about the secondary gradient term?

So, gamma f does not change for the cells whether you are sitting on C_0 or C_1 you calculate this is the same, $\overrightarrow{\mathrm{A_f}}$ dot $\overrightarrow{\mathrm{A_f}}$ does not change the same thing, what about $\overrightarrow{\mathrm{A_f}}$ dot e_ξ? $\overrightarrow{\mathrm{A_f}}$ dot e_ξ dot both of them change signs so, as a result, there is no problem so, this entire thing remains the same that is why, our primary gradient was coming out to be plus sign for both of them.

But what about e_ξ dot e_η? What about this quantity? This we have to see. What about $\frac{\partial\phi}{\partial\eta}$? This quantity would remain the same right. You would calculate from ϕ_a and ϕ_b and calculate the gradient; this is the same for whether you calculate from a cell or from b cell ok.

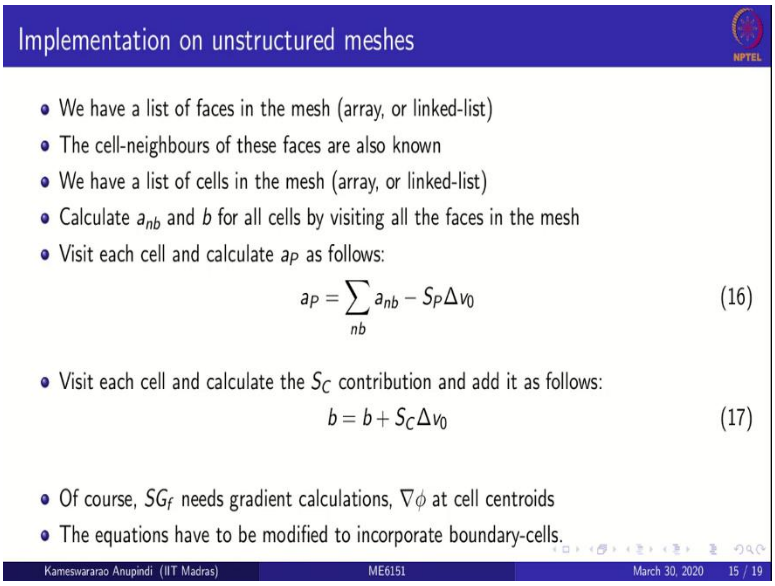
What about e_ξ dot e_η? So, e_ξ dot e_η, when you are calculating from C_0 side for the face f, e_ξ is pointing this way, e_η is this way whereas if you have let us say gone to the next cell and try to calculate secondary gradient for this space, then e_ξ would point in this direction and e_η would still point in the same old direction right because we are assuming that it points in this section.

As a result, what happened to the angle? Earlier, e_ξ dot e_η had this angle here some acute angle, now e_ξ it is on my obtuse angle and then, now, e_ξ dot e_η has this angle right. So, there is a change in angle here by 180 here. So, as a result, e_ξ dot e_η would change sign when you go from C_0 to C_1 because of the assumption that e_η is still pointing in the same direction ok. So, as a result, you have to add it to the b 0 cell and subtract it from the b 1 cell right. Is this clear?

So, when you move, we are trying to calculate f from one cell and assign it to the both cells right. So, as a result, all these terms will remain the same except for e_ξ dot e_η ok, e_ξ dot e_η is flipping because e_ξ is flipping, but e_η is not. As a result, e_ξ dot e_η would give you a sign change which needs to be accommodated in the calculation. So, because SG f is calculated from C_0's perspective, that is why we are subtracting it off here right.

Of course, if you calculate again go back to C_1 and calculate then this will be plus only right that we are not doing here ok. So, that means, the primary gradient can be directly assigned, but the secondary ingredient has to be added to one cell, the b term of one cell and subtract it from the b term of the other cell ok. So, that is what we kind of discussed till now fine.

(Refer Slide Time: 37:51)



Implementation on unstructured meshes

- We have a list of faces in the mesh (array, or linked-list)
- The cell-neighbours of these faces are also known
- We have a list of cells in the mesh (array, or linked-list)
- Calculate a_{nb} and b for all cells by visiting all the faces in the mesh
- Visit each cell and calculate a_P as follows:

$$a_P = \sum_{nb} a_{nb} - S_P \Delta v_0 \tag{16}$$

- Visit each cell and calculate the S_C contribution and add it as follows:

$$b = b + S_C \Delta v_0 \tag{17}$$

- Of course, SG_f needs gradient calculations, $\nabla \phi$ at cell centroids
- The equations have to be modified to incorporate boundary-cells.

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 15 / 19

Then, what we assume is that through the grid generation software or through the cells we create, we have a list of faces in the mesh that we have generated. That means, we know either through an array or through a linked list, what are the faces we have in the mesh.

Now, the cell neighbors for these faces are also maintained. For example, you go to your particular face, you know who are your cells C_0, C_1 and go to another cell, you know is your C_0, C_2 and so on. So, and also, we have a list of cells in the mesh ok; that means, either through an array or a linked list, we maintain list of cells. Now, these cells are formed because of the faces. So, we can go from cells to faces and faces to cells in terms of the connectivity fine alright.

So, having given that we know all these three that we have, then how do we approach in writing a solution algorithm for unstructured measures? Essentially, you visit all the faces and when you visit every face in the mesh, you calculate what is $\mathrm{a_{nb}}$, what is b for the cells right and then, assign it to $\mathrm{a_{nb}}$ for both the cells. And, b once you calculate what is second gradient, you assign it to one cell, add it to one cell, subtract it from the other cell and keep doing this.

Once you finish all the visiting all the faces, you have got a nb calculated for all the cells and b the contribution of secondary gradient is also added for all the cells right that is done. Then, what we do is we will not visit then, we will not go to the faces, we will go to a list of cells. So, we visit each cell, and we calculate $\mathrm{a_P}$ ok. How do when I calculate $\mathrm{a_{nP}}$?

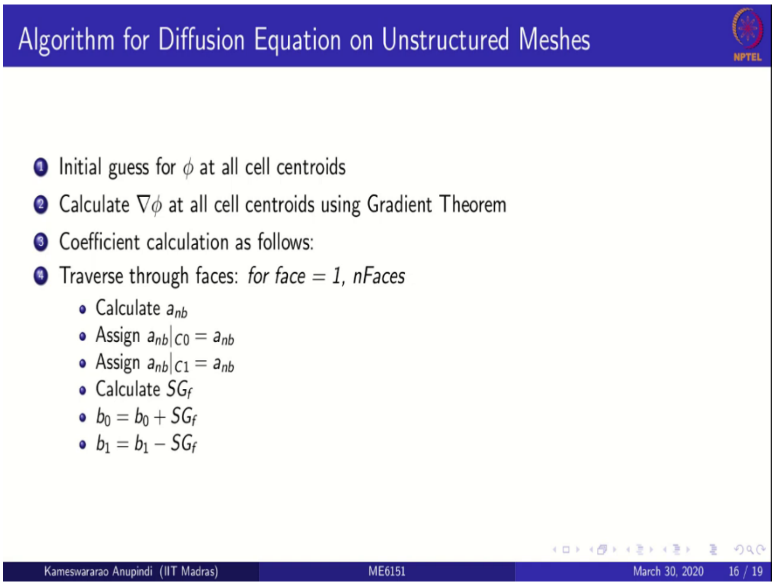
Now, every cell got $\mathrm{a_{nb}}$ is populated. So, I can go simply sum all the $\mathrm{a_{nb}}$'s that cell has and that summation minus $\mathrm{S_P}$ times ΔV_0 will give me what is $\mathrm{a_P}$ right. Of course this has to be modified if you have a boundary condition, a cell that is near the boundary. Because your contributions would change right because you would have if it is a Dirichlet, you would get an extra quantity here and so on that has to be done, we are talking about an interior cell in this particular context alright.

Now, then visit each cell and calculate the contribution of S_C and add it to the b term right. The b term now already contains the secondary gradient component. Now, add S_C times ΔV_0 to the b for every cell ok. Now, we are going cell by cell. So, when we visited the faces, we populated the $\mathrm{a_{nb}}$'s and b's and then, when we are going by cells, we are calculating what is $\mathrm{a_P}$ and what is updating b for every cell right alright. So, till now, there is no duplicacy as this right, we are doing everything once and then using it as much as possible.

Now, of course, the second gradient f needs to needs gradient calculations and $\nabla\phi$ at cell centroids these were somehow already calculated before we go to this step right. Because, you already know what is your calculate, what is $\nabla\phi$ before this step and then using $\nabla\phi$, you have calculated what is SG f and then you populated $\mathrm{a_{nb}}$ and b and so on right.

Now, of course, these equations have to be modified, if you have to incorporate the boundary cells right, if you have be incorporate the Dirichlet boundary condition or Neumann boundary condition, these have to be changed. I think that we understand fine. So, that is the overall algorithm.

(Refer Slide Time: 41:24)



Algorithm for Diffusion Equation on Unstructured Meshes

1. Initial guess for ϕ at all cell centroids
2. Calculate $\nabla\phi$ at all cell centroids using Gradient Theorem
3. Coefficient calculation as follows:
4. Traverse through faces: *for face = 1, nFaces*
   - Calculate a_{nb}
   - Assign $a_{nb}|_{C0} = a_{nb}$
   - Assign $a_{nb}|_{C1} = a_{nb}$
   - Calculate SG_f
   - $b_0 = b_0 + SG_f$
   - $b_1 = b_1 - SG_f$

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 16 / 19

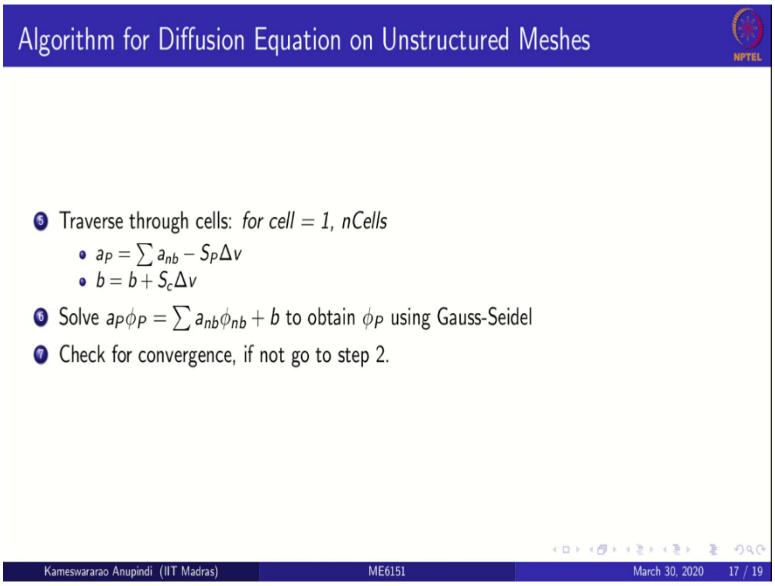
Let us look at the complete algorithm as it stands for solving diffusion equation on an unstructured non-orthogonal mesh ok. So, what do we do? Initially, we start off with phi at the cell centroids right that is the 1st step because if you want to use a an iterative based method, you guess what is phi at the cell centroids and also at the face centroids where you have to apply a Neumann boundary condition right or a mixed boundary condition. So, an 1st step is to guess phi at the cell centroids.

Now, the 2nd step is basically calculate a gradient of the phi. Why do we do this thing uh? Because we need this in the secondary gradients, we need this in all other terms that might go into b right. So, using either gradient theorem or using least squares method, you calculate what is $\nabla\phi$ so, once you with the initial guess that you have alright. Then, we calculate all the coefficients.

So, for calculating the coefficients, we first traverse through the faces. So, we have a list of faces. So, if I use f face here as my index i, so, I am going from face 1, 2, 3 and so on to n faces which is basically the total number of faces I have in a particular mesh ok. So, that is what we do, we traverse through the faces one by one.

And when we are at the face f, we calculate a_{nb} that is we calculate A_f and assign it to both the cells C_0 and C_1 as it is, then you calculate what is secondary gradient and you add it to the C_0 cell that is b_0 and you subtract it from the b_1 cell, b_1 equals b_1 minus second ingredient of f, b_0 equals b_0 plus secondary gradient of f. So, we do this thing this is basically calculating the coefficients and filling them right that is done. Then, we are done visiting all the faces.

(Refer Slide Time: 43:22)



Algorithm for Diffusion Equation on Unstructured Meshes

5. Traverse through cells: *for cell = 1, nCells*
   - $a_P = \sum a_{nb} - S_P \Delta v$
   - $b = b + S_c \Delta v$
6. Solve $a_P \phi_P = \sum a_{nb} \phi_{nb} + b$ to obtain ϕ_P using Gauss-Seidel
7. Check for convergence, if not go to step 2.

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 17 / 19

What we do next? Next is to look at cells right. So, we go to each and every cell right, traverse through the cells for cell is my index here some kind of i or j 1, 2, 3 and so on so, cell go from first cell, second cell, third cell and all the way to n cells that you have and your here is where we are calculating the central coefficient that is a_P, a_P equals sigma $\mathrm{a_{nb}}$ which were already populated in a visit to the cells through the faces minus $\mathrm{S_P}$ times ΔV right this is basically the source term that is going into a_P.

Now, of course, we have to also add the b term that is b plus $S_C\ \Delta V$ right this is what we do for the b term and a_P term, when we are visiting cells alright; that means, now in a way by calculating $\mathrm{a_{nb}}$, b and a_P, we have actually formed this equation in a virtual manner right.

We have nowhere written this equation as it is, but we have updated these values; that means, we have this everything that we want in this particular equation to be solved right that means we have $a_P \phi_P = \sum \mathrm{a_{nb}} \phi_{nb} + b$ all the coefficients are non known and also the phi and b's are known from the guess values.

So, we are now ready to solve this equation for ϕ_P for every cell ah for this particular iteration right. Essentially, we can use Gauss-Seidel or Gauss-Seidel successive or relaxation or under relaxation and then, calculate what is ϕ_P for the entire cells right.

Now, once you do that, you check for convergence. Check for convergence meaning you have two convergences especially, here you converge everything, you solve for ϕ_P not changing between iterations now, that ϕ_P you check it with the whatever is the guest value in step 7 right.

Now, this of course, would not be the same so, it would so, although the Gauss-Seidel has converged this would be different because the guess value will be different from this converged value, then if it is not the same then you go back to step 2 right, step 2 is what? Then, with the new values that you got for phi, you calculate $\nabla\phi$ and then, follow down the same process and then again solve for Gauss-Seidel, obtain new phi values at all the cell centroids and move on ok.

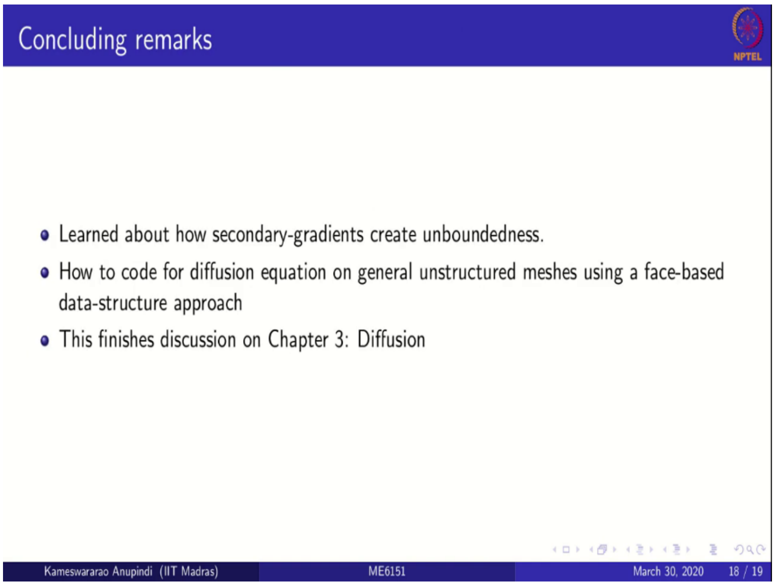
Now, you see here there are two iterations what we call as inner iterations which are basically for solution of the Gauss-Seidel right. This is let us say some 1000 iterations, these are all the inner iterations. Now, once you come out, you are doing this if not go to step 2, this is basically the outer iteration right which is basically telling you that you have to keep doing this for several iterations until you converge to some value. Now, we are talking about a steady diffusion equation only right.

But we still have two iterations that is because we are using an iterative method for the solution. The other one is because we have a non-orthogonal mesh. Now, these outer iterations are required; are required if you have non-linearity in your problem right. If you have non-linearity, you need these order iterations anyway that is one particular case. If you have a linear problem and if you have a orthogonal mesh, then you do not have to have this outer iterations right because everything you are solving for is correct.

But if you have still a linear problem, but if you have a non-orthogonal mesh, then you still have to have outer iterations because the secondary gradients are calculated in a deferred correction method right, we are calculating the secondary gradients the $\nabla\phi$ is calculated with the guess value. So, as a result, even if you have a linear problem, if you have a non-orthogonal mesh then you would still require outer iterations ok.

But if you have a linear problem on an orthogonal mesh, then you do not need the outer iterations is that clear need for the outer iterations. So, the need for the outer iterations is even for steady state problems either non-linearity or non-orthogonality would demand the outer iterations fine alright. So, that is how you would code a diffusion equation on an unstructured mesh that is either orthogonal or a non-orthogonal ok. So, this could be a good method to implement yeah so, for your project maybe let us see.

(Refer Slide Time: 48:05)



Concluding remarks

- Learned about how secondary-gradients create unboundedness.
- How to code for diffusion equation on general unstructured meshes using a face-based data-structure approach
- This finishes discussion on Chapter 3: Diffusion

Kameswararao Anupindi (IIT Madras) ME6151 March 30, 2020 18 / 19

So, concluding remarks are basically in this lecture, we learned about how secondary gradients create unboundedness and then, how do you program for diffusion equation on general unstructured meshes using a face-based and a cell-based data-structure approach ok.

So, we have answered these two questions in today‘s lecture alright and this kind of finishes our chapter 3 on the diffusion equation ok. So, we have discussed all the structured, unstructured meshes, steady, unsteady diffusion equation and we have looked at the stability and the accuracy all these things for the diffusion equation alone ok.

So, in the next lecture, we are going to look at, we are going to start the next chapter that is chapter 4 that is for convection, convection and convection diffusion. So, that will be our discussion for next lecture alright. So, I am going to stop here, thank you. If you have questions write back to me, we will, we can discuss them ok.

Thank you.