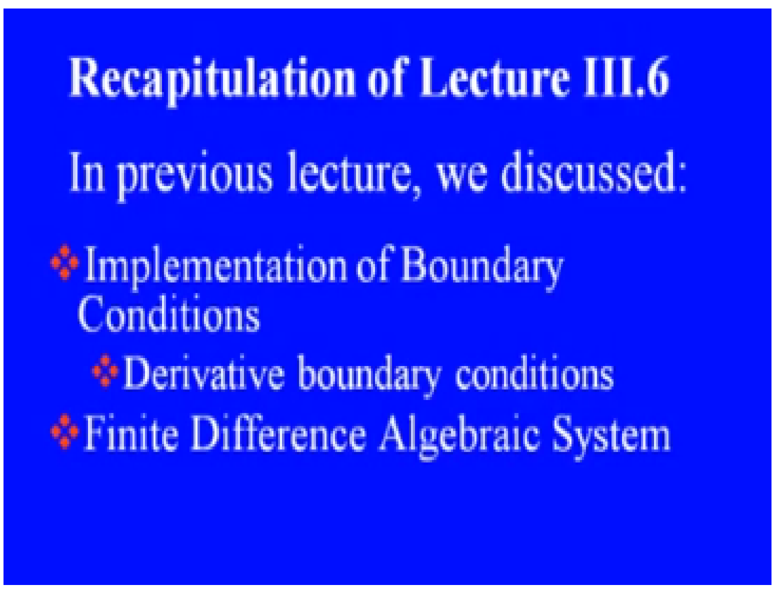
Computational Fluid Dynamics
Dr. Krishna M. Singh
Department of Mechanical and Industrial Engineering
Indian Institute of Technology – Roorkee

Lecture – 16
Applications of FDM to Scalar Transport Problems-1

Welcome back to module 3, we are now in the concluding like on finite difference method wherein we would consider application of FDM to few scalar transport problems and we will also discuss few computational aspects, that is to say, you take up the case that if you want to implement finite difference method using a computer programming language which of the aspects, which we are to consider.

So, these were 2 things which we are going to do in this lecture and possibly in the next lecture as well.

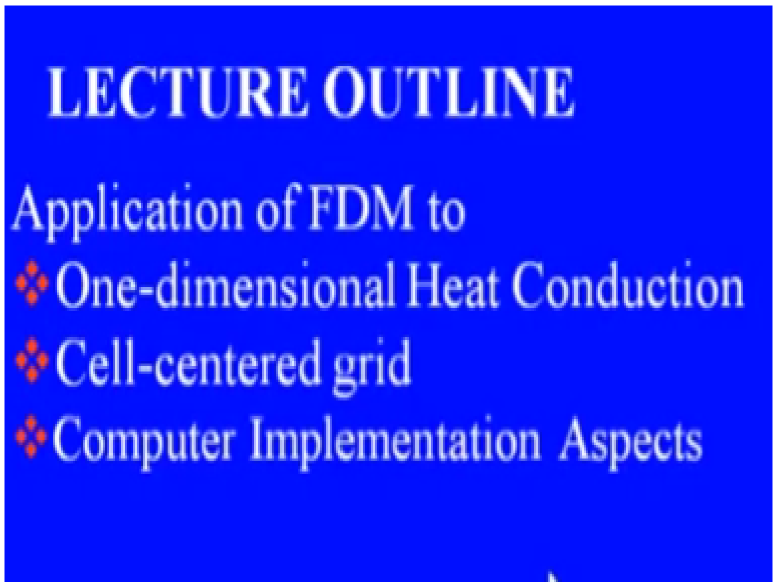
(Refer Slide Time: 01:33)



So, our model outline, we already covered the shaded parts. Today, we are going to focus on applications of FDM to scalar transport problems. Before you proceed further, let us have a recapitulation of what we did in the previous lecture. We discussed implementation of boundary conditions specifically the ones which involved derivative boundary conditions wherein, we need to come up with suitable finite difference approximations of the derivatives.

And then we discussed features of finite difference in algebraic system which we obtain from the discretisation of work continuum problem, so we discussed in detail at nodal discrete algebraic equations and what we mean by computational molecule or a stencil in the context of finite difference discretisation and we briefly looked at indexing and storage aspects as well. In today's lecture, we are going to focus on applications of finite difference method to scalar transport problems.

(Refer Slide Time: 02:41)

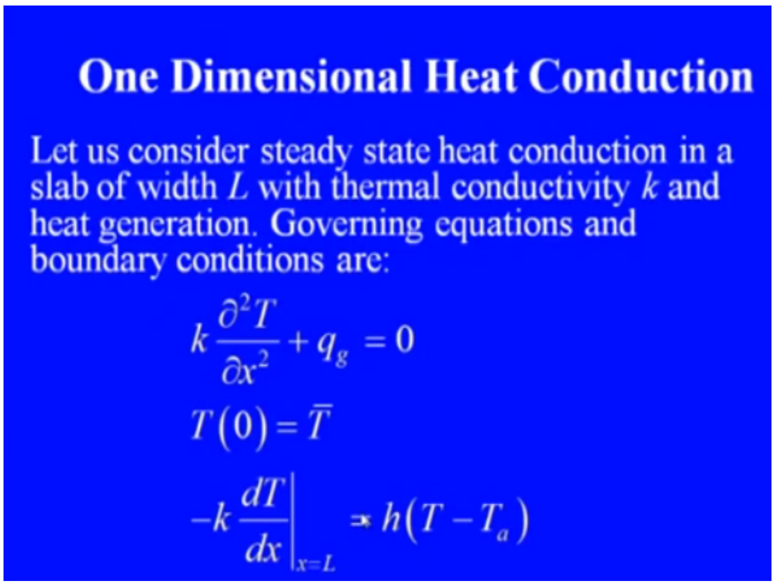


We would initially focus on simple one D heat conduction problems and later if time permits will take up 2 dimensional heat conduction and advection diffusion problems, so outline of lecture we would try and apply finite difference method to one dimensional heat conduction. Now, today I would introduce heat and their concept that is related to the choice of our grid, which we called cell centered grid.

And the cell centered grid is specifically of great importance in the context of Navier-Stokes equations were in cell centered grids are primarily used, the nodes of the cell centered; they are used for the discretisation of all scalar quantities and the nodes on the surfaces of control volumes, they are used for velocity components. So, keeping that in view, now let us have a look at a cell centered discretisation apply to where, one dimension heat conduction you would see if it hoppers any advantages.

Then, we will briefly discuss that so for whatever algorithm which we have developed for one dimensional heat conduction. If you want to implement, we want to write a computer program, so what are things we need to consider and then if time permits, we will move on to application of finite difference method to 2 dimension heat conduction and possibly advection diffusion problem.

(Refer Slide Time: 04:01)

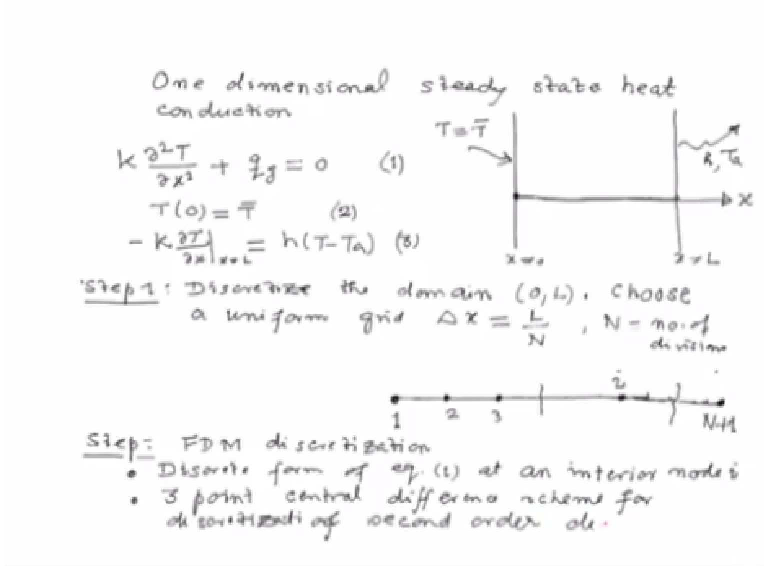


If not, we will cover these 2 topics in our lecture. Now, let us come to a first application; one dimensional heat conduction, so let us consider steady state heat conduction in a slab width L with thermal conductivity k and uniform volumetric heat generation. So, for this problem we already derived our governing sequence earlier in module 2, just rewrite or let us rewrite our governing equations.

Governing equations and boundary conditions in the case based on the Fourier's law are, k del 2 T/del x square + qg =0, wherein this case, thermal conductivity which has been assumed to be constant and T is the temperature and q and g are volumetric heat generation. Now, like the left and that is x =0. We have got temperature specified see, T at 0 = T bar, where T bar is specified temperature.

Next, let us suppose the right end, the wall or slab that is exposed to the moment and there is a heat loss due to conviction, so in that case our convective boundary condition becomes –kdT/dx at x = L, this edge of a heat flux, this would be equal to h times T–Ta, where T is the temperature at the wall and T is ambient temperature, h is our convective heat transfer coefficient. So, now let us derive the finite difference approximations for this problem.

(Refer Slide Time: 05:48)



So, for derivations, let us move on to a board, so we dealing with one dimensional steady state heat conduction, for the sake of reference, let us rewrite our equations on the space as well, so we had our governing equation given by k del 2 T over del x square + qg = 0 zero, list number this as equation 1, then we get 2 boundary conditions T at zero that was given by T bar, just call this equation 2.

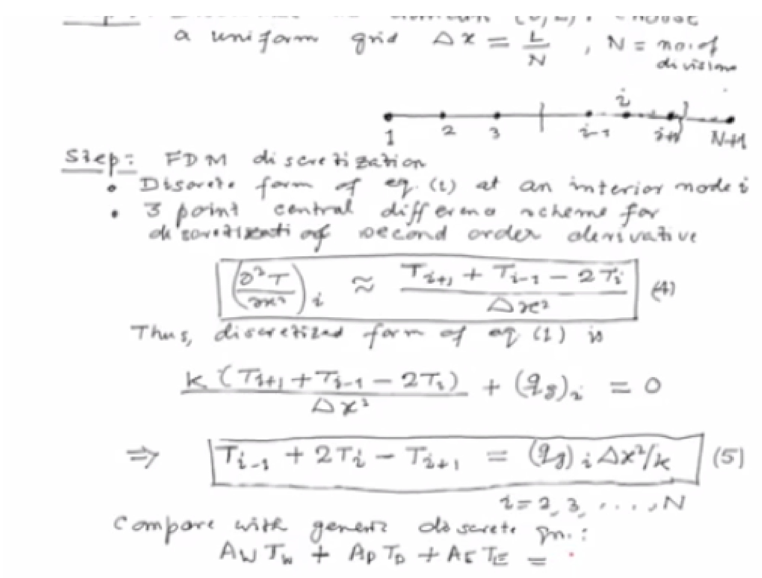
And the third boundary condition was given as –k del T/ del x at x = L h*T–Tn, let us call it as third boundary condition, so physical problems something like this, at sloped slab, the left ends we fix our coordinate origin. This is our x direction, so we have got temperature specified here; T = T bar and we have got convection to a medium which is winding kind of temperature Ta and convective heat transfer coefficient is given by h.

Now, a finite difference applications list; discretised dimension that is the first aspect, so first task is or step one; discretise the domain since it is one dimensional problem, we need to consider only discretisation along the line, so domain would now be given by 0 to L and let us choose for the sake of simplicity of our discussions today. Let choose a uniform grid delta x which is given by; if we use N deviations of our computational domain, the total length is L.

So, delta x would become L/N, where N is number of divisions or we can also call it number of cells, so, let us; now let us draw our one dimensional grid with the proper grid indices, so the leftmost node, we will; that is x = 0, we would number it by 1, 2, 3 and so on, then the generic node i and around that we will have nodes i -1, i +1 and so on and towards the end we have the last node.

Since we have got N divisions, the last node would be numbered as N+1, our next step would be finite difference discretisation that is obtain discrete form FDM discretisation, it will consist of obtaining discrete form for our governing equation and boundary conditions, so now first let us obtain discrete form for a governing equation; form of equation one at an interior node, while interior node meaning any node other than node 1 and N +1.

(Refer Slide Time: 11:30)



Both of which are in the boundary of the domain, node i and let us use 3 point central differences scheme for discretisation of work derivative, discretisation of second order derivative that is to say this or del 2 T over del x square at the interior node I, we are going to approximate it in terms of the nodal values at the grid point i-1 and i +1, so central difference formula which we learned earlier this is simply, T of i+1 + T of i-1 -2Ti / delta x square on our uniform grid.

So, this is the approximation of the derivative which is involved in our governing equation one, let us call this equation as 4, so if you substitute 4 into in our equation 1, so then we get discretised form; discretised form of equation one is k times T of i +1+ T of i -1-2 Ti / delta x square + our qg term evaluated at node, I this would be equal to 0, let us rearrange this equation, so we can multiply it by delta x square/ k.

Keep this unknown temperature terms on left hand side and the known source term lets sifted on the right hand side, so if you do that, we will get T of i -1+2 Ti-T of i +1 = qg of i delta x square/ k. So, this equation holds good for all nodes; for all nodes i= 2, 3 up to N, so this is the discrete form of our governing equation which has been obtained using central differences scheme. Now, let us compare it with generic equation which we wrote earlier.

(Refer Slide Time: 15:20)

- 3 point central difference scheme for discretisation of second order derivative

$$\boxed{\left(\frac{\partial^2 T}{\partial x^2}\right)_i \approx \frac{T_{i+1}+T_{i-1}-2T_i}{\Delta x^2}} \quad (4)$$

Thus, discretized form of eq (1) is

$$\frac{k\,(T_{i+1}+T_{i-1}-2T_i)}{\Delta x^2} + (q_g)_i = 0$$

$$\Rightarrow \boxed{T_{i-1}+2T_i-T_{i+1} = (q_g)_i\,\Delta x^2/k} \quad (5)$$

$i = 2, 3, \ldots, N$

Compare with generic discrete eqn.:

$$A_W T_W + A_P T_P + A_E T_E = q_P \quad (6)$$

W P E
i-1 i i+1

$A_P = 2$, $A_E = A_W = -1$

$q_P = (q_g\,\Delta x^2)/k$

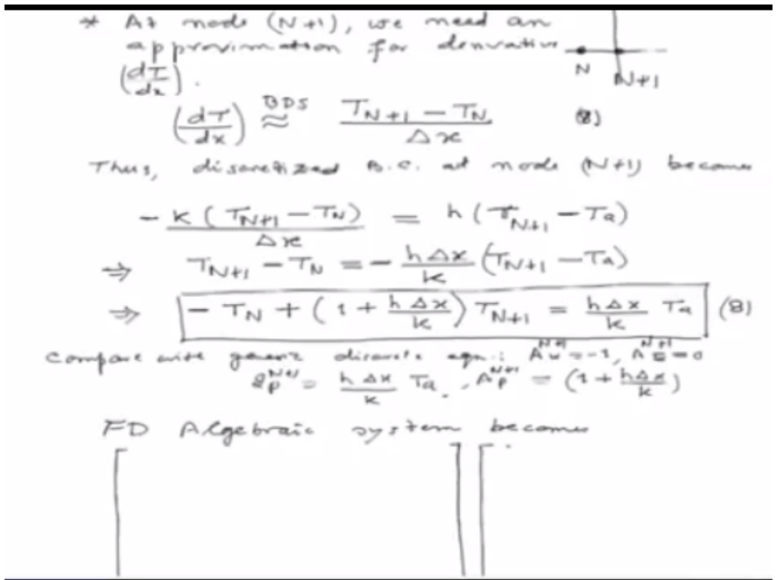
Generic discrete equation, so compare with generic discrete equation which we have written as AW, TW + AP TP + AE TE = QP, now let us recollect the correspondence between this compass notation and our indices, so this P stands for our node I, E stands for the index i +1 and W stands for index i -1. So, if we compare equations 5 and 6, we can easily see that our AP = 2 AE and AW the both = -1 and QP = qg delta x square/ k, qg evaluated at grid point i.

(Refer Slide Time: 16:39)

So, now you have been able to discretised one part that is our differential equation. The next part is we have to look at the boundary conditions, so now let us discretised the boundary conditions, for discrete form for boundary conditions at node 1, what we have? T, so node one that is; which corresponds to T=0, temperature is specified as T bar, so the equation becomes very simple here. Hence discrete equation for node 1 is; T1 = T bar, just call this equation as 6.

So, if you compare it with the generic equation what we have for this? Thus our AP for node 1 is equal to 1 AW of and AE for node 1, they are both 0 and QP for node 1 that is T bar. Next let us move on to the rightmost boundary which corresponds to our node N+1, that node N +1, we need an approximation for an derivative dT/dx and the simplest approximations which you are going to choose is; our 2 point backward difference scheme which will involve values at N +1 and node N.

(Refer Slide Time: 20:24)



And now let us approximate this dT/dx using BDS, so we can write this is as TN +1-TN / delta x, substitute this approximations to derive in our convective boundary condition, thus discretised BC at node N +1 becomes -k times TN +1-TN / delta x = h times, on the right hand side T, that T corresponds to TN +1 – Ta. Now, let us rearrange these terms a little bit, so we can write this as T N +1-TN = -h delta x/k, we transfer this delta x and k to the right hand side.

We also transfer the negative sign, so we get -h delta x/ k which multiplies TN +1 – Ta, transfer the unknown variables TN +1 to the left hand side, sorry let us rewrite this equation as – TN and collect all the coefficients of TN+1 together, so we get 1 + h delta x/ k TN +1 = h delta x/ k times Ta, so let us call this equation as 8, so if we compare with generic discrete equation or generic discrete equation, so we get AW = -1 AE = 0.

AP, let us put this N+1 as superscript, so AP N +1 this is equal to 1+ h delta x/k and our qp N +1 that is h delta x/ k times Ta. So, now you got all the equations at all nodes in discrete form, so our final algebraic equation can be written as; finite difference algebraic system becomes; now let us write it out in the matrix form, for node 1 what we had? The coefficient of the matrix; let us put the unknown terms; T1, T2 and so on.

(Refer Slide Time: 23:36)

$$-\frac{k\,(T_{N+1}-T_N)}{\Delta x} = h\,(T_{N+1}-T_a)$$

$$\Rightarrow \quad T_{N+1}-T_N = -\frac{h\Delta x}{k}\,(T_{N+1}-T_a)$$

$$\Rightarrow \quad \boxed{-T_N + \left(1+\frac{h\Delta x}{k}\right)T_{N+1} = \frac{h\Delta x}{k}\,T_a} \quad (8)$$

Compare with general discrete eqn.: $A_W^{N+1} = -1$, $A_E^{N+1} = 0$

$$Q_P^{N+1} = \frac{h\Delta x}{k}\,T_a, \quad A_P^{N+1} = \left(1+\frac{h\Delta x}{k}\right)$$

FD Algebraic system becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ & & \ddots & \ddots & \ddots & & & 0 & \\ & O_2 & & \ddots & \ddots & \ddots & & & \\ & & & & & -1 & 2 & -1 & \\ & & & & & & 0 & -1 & A_P^{N+1} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_i \\ T_{i+1} \\ \vdots \\ T_{N+1} \end{bmatrix} = \begin{bmatrix} \bar{T} \\ Q_P^2 \\ Q_P^3 \\ \vdots \\ \vdots \\ Q_P^i \\ \frac{h\Delta x}{k}T_a \end{bmatrix}$$

This Ti, Ti +1 and T of N +1, on the right hand side you get, at the first equation that T1 = T bar, so Q P1 that was equal to T bar, for the second equation onwards we had this term, let me put it as QPI, the expression for QPI, we have already derived earlier, the previous space, so it will continues as QPI for all the terms from 2 to N and for TN+1 we had the last term, let us we write h delta x/ k times Ta. So, this completes our right hand side.

Now, let us come to the left hand side, in the first row the only nonzero coefficient that corresponds to the one which multiplies T1, so this main diagonal, we have got one, rest are all zeros. For i=2, we have got AW that was equal to -1, the main diagonal that is AP with corresponds to AP this becomes 2, AE was -1, remaining terms simply zeros. The same holds good for 3rd term; -1, 2, -1, 0, 0, 0, 0 and so on, so we can continue in this way. So, what we will find is that we have got nonzero terms only on the main diagonals.

(Refer Slide Time: 26:33)

1 - D Heat Conduction
FD Algebraic system

(1) It is sparse
⇒ It is a tri-diagonal system.
⇒ It can solved very efficiently using TDMA (Tri-diagonal matrix algorithm)

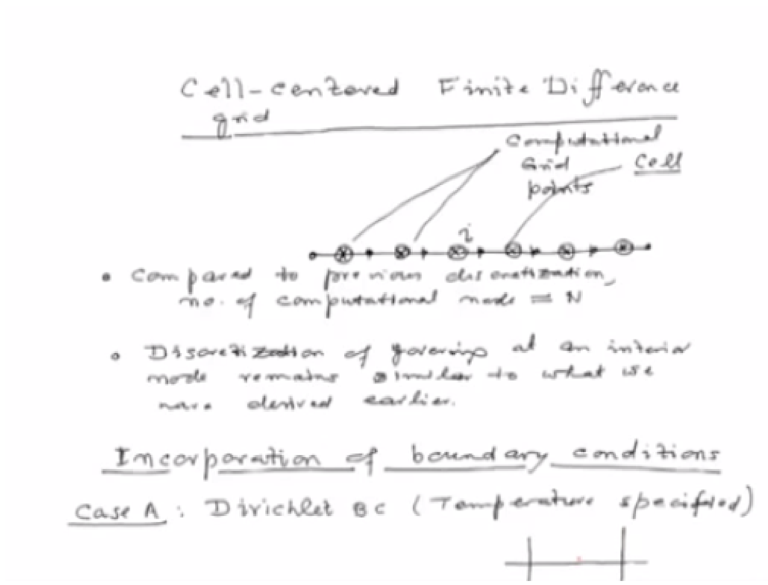
(2) Matrix [A] is not-symmetric.

And remaining are all zeros -1, 2, -1, now let us write the last equation 0, -1 and simply write this as AP N+1 because this is a bit more involved, we have written that earlier. So, if you look at this equation, look at the system matrix. System matrix is sparse that is one characteristic, the other part is; what are the characteristics the system which we have got? So, we have got this one D heat conduction, this is the problem which we have been discussing.

So, the finite difference algebraic system; 2 characteristic we can note; the first one is, it this sparse and this sparsely pattern is very clear, what is that sparsely pattern? It is a tri diagonal system, so the next module when we discussed solution of algebraic recently we will come across or we will discuss one very important and very elegant algorithm which is called TDMA algorithm that is we can solve this system, so it can be solved very efficiently using the TDMA.

This TDMA is an acronym for tri diagonal matrix algorithm and some people also call it as Thomas algorithm and their observation which we can make is; it is not symmetric; so matrix A is not symmetric though here the symmetry is really not of much importance since, the simple one dimension heat conduction for tri diagonal structure and this TDMA algorithm does not care whether the matrix is symmetric or not.

(Refer Slide Time: 29:14)



So, this is just an observation that in this case, we have got a metrics, which is not symmetric by using our normal finite difference grid. So, next time we are going to introduce you a small variant of what we have just discussed and that variant is very important for few problems. So, we will introduce what we call cell centered; cell centered finite difference grid, so let me introduce it using our one D problem.

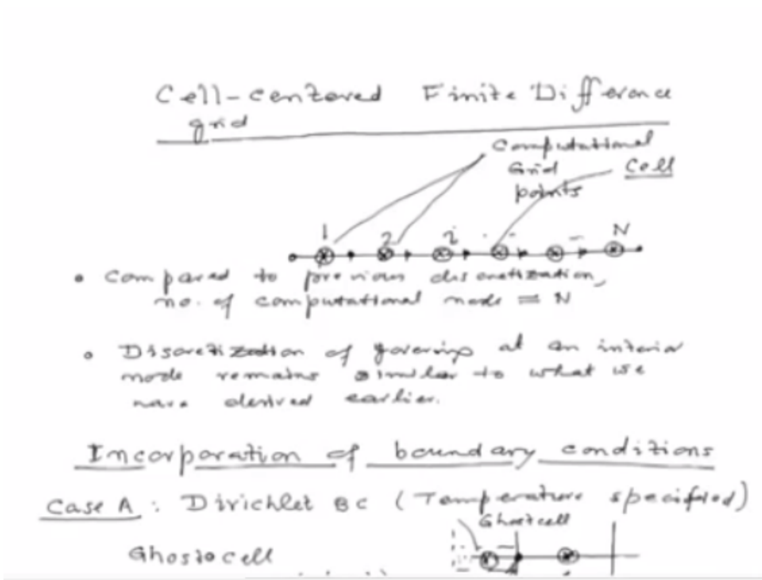
So, one dimension, what we have? It was simple one dimensional domain and for discretisation, we divided into; let us say N equal segments, the segments could be unequalled in really matter. Now, previously what we had done; these were the node that is the points which mark our divisions, those were the ones which we have taken as our grid points. Now, let us make it a slight departure, instead of treating these, dividing points such our computational nodes.

Now, let us take the points which are in the centre of these so called divisions, each of these divisions we can also call them; let us call them as a cell and our computational node, which are now put with the field circle, these are our computational grid points okay, so what would be the difference compared to the previous arrangement, we will have exactly as many nodes, as many their divisions we have made somewhat compared to previous discretisation; compared to previous discretisation is number of computational nodes is one or less, this is now equal to one.

Earlier, we had 1 to N +1 node. Now, as far as discretisation of 4 interior nodes is concerned that would remain unchanged, so discretisation of governing equation and at interior node, this is call as interior node I, that would be once again the same and remains similar to what we had derived earlier, but now there would be an important difference, how do we incorporate the boundary conditions?

Now, we have them in the position of the known temperature boundary condition would require some in general, we have to make certain modifications. So, let us come to the boundary nodes that is; that is our incorporation of boundary conditions, so we will take 2 cases; one of our temperature specified, so case A, which we call Dirichlet BC at this temperature specified, how to tackle this case?

(Refer Slide Time: 34:15)

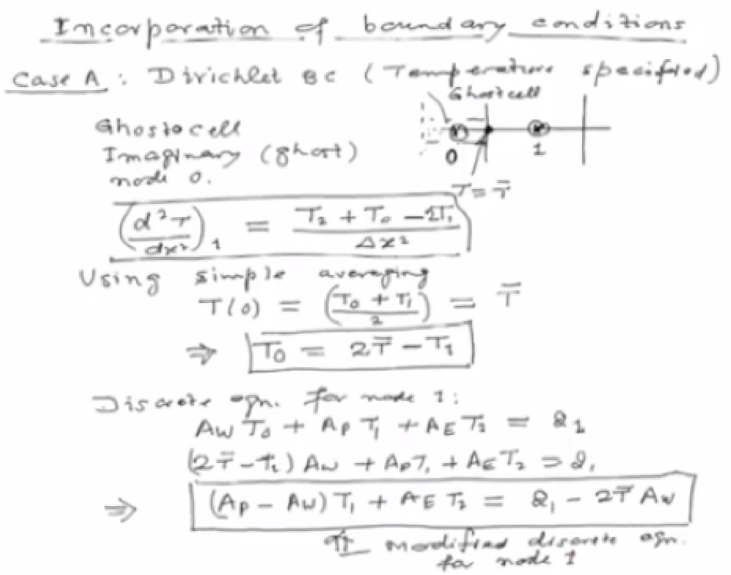


Now, suppose now we are dealing with the temperature specified at node 1 as we have been discussed in our specified problem, so now let us draw the cell; the first cell, so our computational node is setting here at the centre, the temperature is specified this boundary node, this were we have got, T = T bar, okay. What do we do now? We introduce a concept of ghost node or ghost cell.

Ghost cell is something which is present, exist in a physical domain, it is just an imaginary extensions, so let us extend this to the left hand side and this dotted portion, this becomes our so called ghost cell. So, we have now an imaginary computational node which we are going to use only for the sake of formulation, it will not come in our problem solution and list number it as 0, okay.

So, you go to an imaginary node or other recorded at the ghost node 0, now this would appear in our central difference approximation at node 1, so if we look at the approximation of our term d2/dx square at node 1, this would be given by T2 + T0 –T1/ sorry twice of T1/ delta x square, okay now this T0 is really not a part of a solution, we have just introduced it to handle the boundary condition, okay.

(Refer Slide Time: 36:55)



So, in this case while using cell centered thing, we would use our usual central difference approximation for all grid nodes 1 to N, okay and we will annual modify the creases corresponding to that nodes on the two sides by which will incorporate boundary conditions at node 1 and node N. So, now in this equation for the node 1, we need value of T0, now let us a very simple averaging concept; using simple averaging.

T at 0 zero, this can be expressed as an average of the temperature at T0 and T1, okay and our T0 was specified boundary condition okay, so now you say that we can write this T0 as twice of T bar – T1, so in the discrete equation for node 1, we need to substitute this value for T0 and that will lead to 2 modifications, this T0 is linked for to discretised equation for node 1; so discrete equation for node 1, this one was our AW T0 + AP T1 + AE T2 = Q at point 1. So, what changes we need to make, if you substitute for T0 that AW thinks will vanish.

And the contribution we will have here 2T bar-T1 * AW + AP T1+ AE T 2 = Q1 or rearrange it, so we can write it as AP-AW times T1 + AE times T2 = Q1 -2T bar AW, so now this becomes our modified discrete equation for node 1, for node 1. Now, we need to adopt a similar concept for introduction of the boundary condition at the last node, okay and then we have got a derivative in our boundary conditions.

So, what we would use instead of using a backward difference approximation which is only first order accurate, which we had used previously. Now, we would use a central difference approximation for the derivative at the boundary node once when we introduces the concept of a ghost cell or ghost node in terms of the valued ghost node, we will write down our derivative, obtain the derivative.

And then substitute the value of the temperature value at the ghost node in terms of what we obtain for boundary conditions into the discrete equation for node in which we had obtain by using central difference approximation for the interior node N, so we will again get a modified equation okay and that modified equation cannot be coupled with the rest of the equations. Now, these things we are going to discuss in the next lecture.

And then we would compare the solutions or other system matrix which we get in both the cases, the one which we get from cell centered, approximate finite difference approximation or vertex centered which we normally use in finite difference and we will notice, there is one very important difference. But for the time being less keep your excitement contained for the next lecture.

So, the next lecture apart from it, we would also try and give one numerical example, we will solve a numerical problem and say which of these approximation; approaches or vertex based approach which your normal finite difference approach or this cell centered finite difference discretisation, which one is more accurate. We will take one problem for which we know what the analytical solution is, so we can compare the two rather easily and then we will move on to few more applications and we will discuss the computer implementation aspects also in the next lecture.