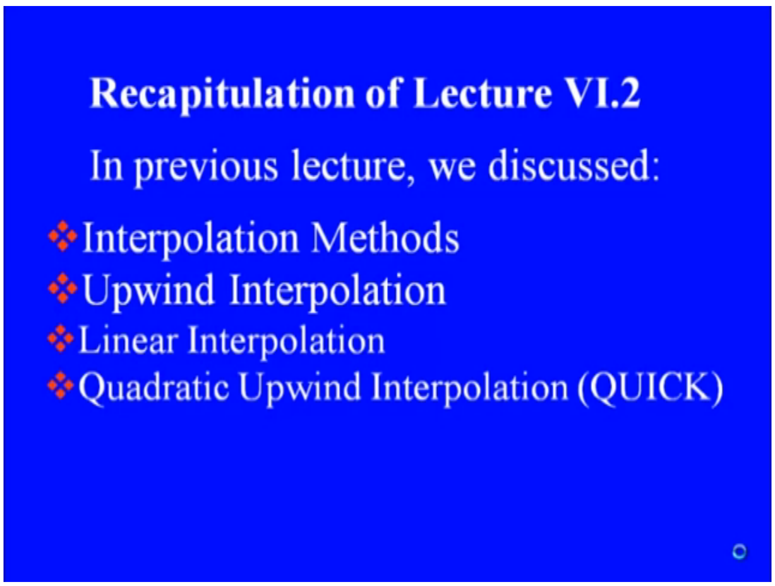
Computational Fluid Dynamics
Dr. Krishna M. Singh
Department of Mechanical and Industrial Engineering
Indian Institute of Technology - Roorkee

Lecture - 30
Application of FVM to Scalar Transport

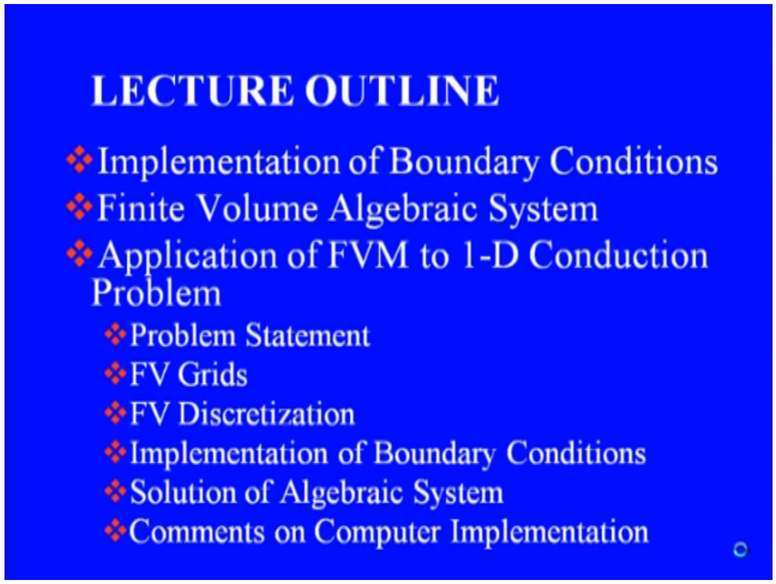
Welcome to the third lecture in module 06 on Finite Volume Method. The previous lecturers we had considered the basic introduction to finite volume that is approximation of finite volume integrals and interpolations schemes. In this lecture we would focus on application of Finite Volume Method to Scalar Transport Problems. So let us have a recap of second lecture which we had earlier.

(Refer Slide Time: 00:55)



So we discussed interpolation matters in particular we had a look at upwind interpolation, linear interpolation and quadratic upwind interpolation method. We also briefly mentioned about the interpolation methods, now in this lecture we are going to focus on application of what we learnt about finite volume formulation to scalar transport problem, and as a model example we would take 1 dimensional steady state heat conduction problem.

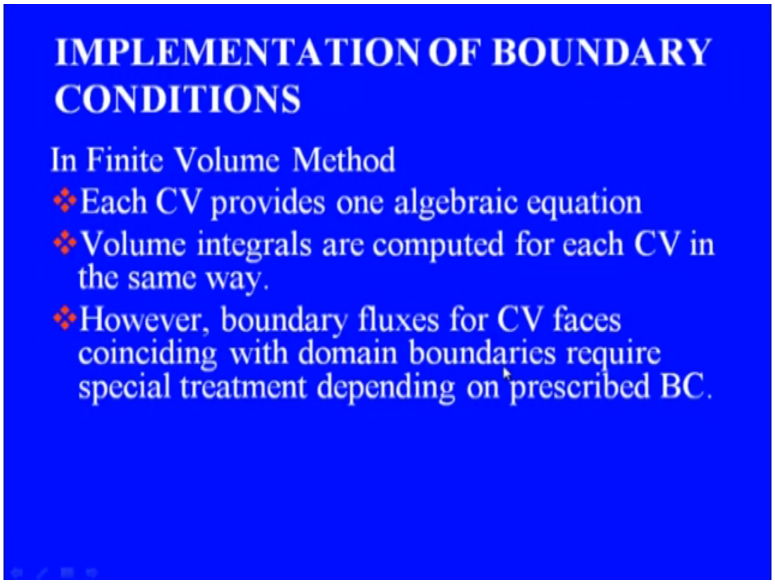
(Refer Slide Time: 01:27)



So outline we will first have a brief look at the implementation of boundary conditions, and we will just review the characteristic of finite volume algebraic system on a structured grid, and then we will have detailed look at the derivation and application of the method to 1-D heat conduction problem, so we will have looked at problems statement finite volume grids which we can use for this problem, we will discuss the discretization using finite volume method.

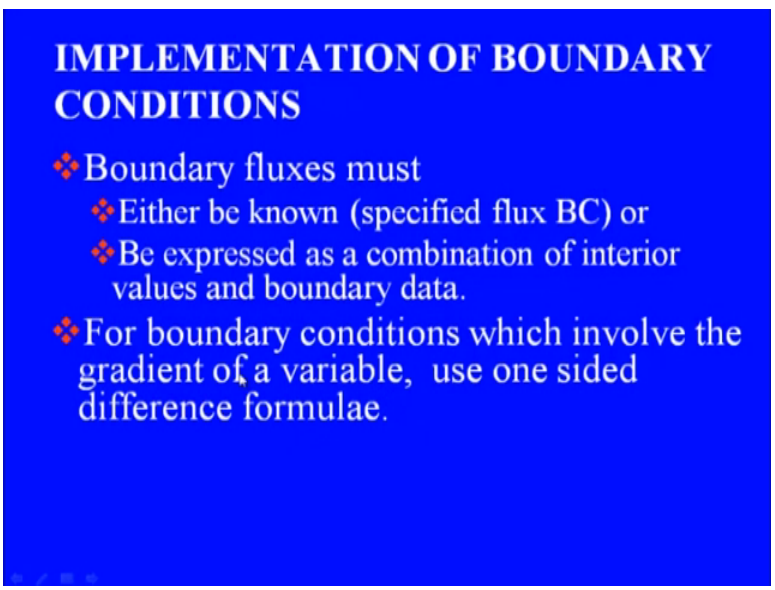
And how do we implement boundary conditions with a different grid choices, and then we will have a brief look at the solution of algebraic system and computer implementation aspects. Now let us have a look at how do we implement boundary conditions in finite volume method the things are fairly similar to the way we did the finite differences, now in case of finite volume for each control volume we get one algebraic equation.

(Refer Slide Time: 02:28)



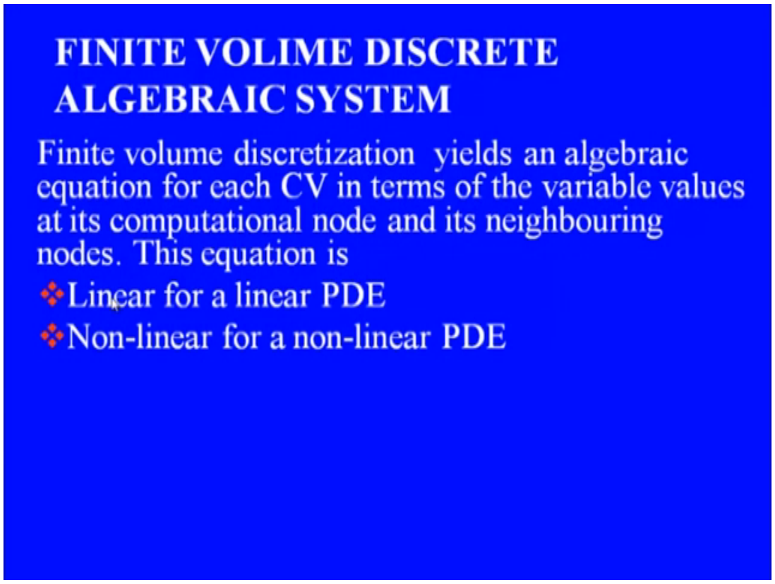
And the volume integrals are computed for each control volume with that is in the interior or close to the boundary in exactly the same way, the boundary conditions affect only the computations or boundary fluxes for the CV faces which coincide with the domain boundaries, and these require special treatment in depending on the prescribed boundary conditions.

(Refer Slide Time: 03:01)



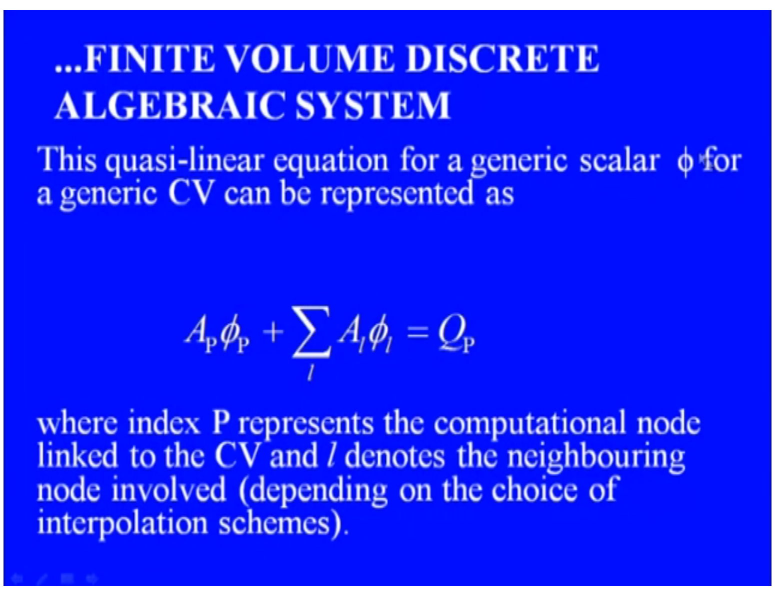
And these boundary fluxes must either be known as a specified flux boundary condition or they can be expressed as a combination of interior values and boundary data, and if we have a boundary conditions which involves the gradient of a variable we can use one sided difference formula in exactly the same manner as well it in a finite difference method. Now a brief look at the algebraic system which we would get when we apply finite volume discretization.

(Refer Slide Time: 03:28)



So discretization using finite volume it is an algebraic equation for each control volume in terms of the variable values at its computational node, which would most likely be at the centroid of this central volume and its neighboring nodes, and this equation would be a linear equation if a problem is linear or it would be non-linear discrete equation if it is a nonlinear problem. Now let us have a look at the way we can represent it.

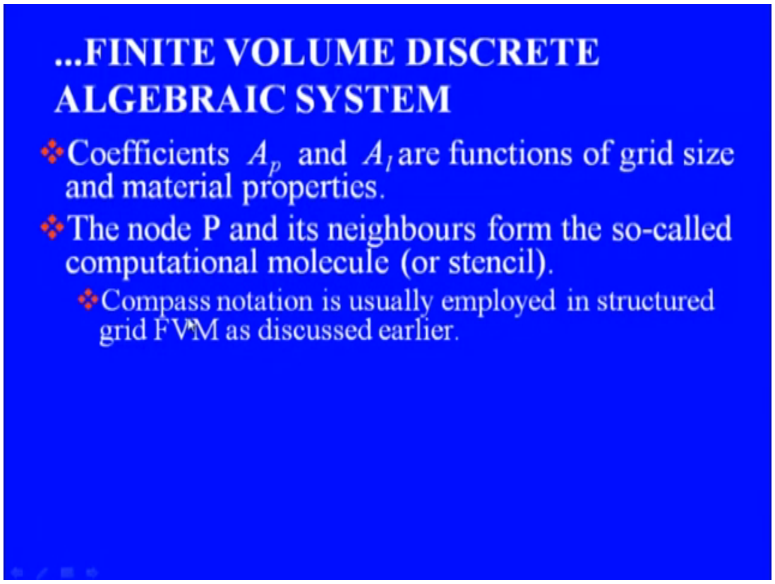
(Refer Slide Time: 04:00)



If we went for a non-linear problem we can represent it as a quasi-linear equation and this quasi-linear equation for a generic scalar quantity phi for a generic control volume can be represented as A capital P*phi capital P+ summation over l Al phi l=QP, now here this index P that represents the computational node which is linked to the CV that is which represents centroid of the control volume and l represents the neighboring computational nodes which are involved.

Now how many number of computational nodes would be involved that would depend on the choice of interpolation screens and the choices for if we have used for the evaluation of surface and volume integrals, in QP would have dumped all the terms which pertain to what we call the load vector which might come from the sources or from the known boundary conditions. Now this is equation which we would get for all the control volumes, this equation will be modified for the control volume which are close to the boundary.

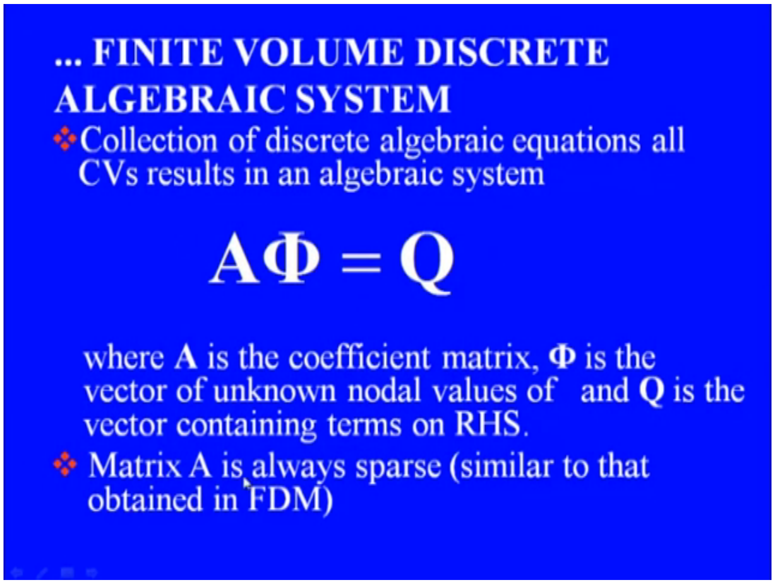
(Refer Slide Time: 05:08)



And if you look at all these one's, now please note that these coefficient Ap and Al they are functions of the grid size that what is size of finite volume and the material properties in similar to the way we had called this P and its neighboring nodes that they form a computational molecule if we have used a structured grid in finite volume analysis, the same technology is also used in finite volume context.

And with these structured grids we would use compass notation, we have already seen it in the first lecture of on this module that how do we represent various nodes and faces on any structured grid in finite volume context.

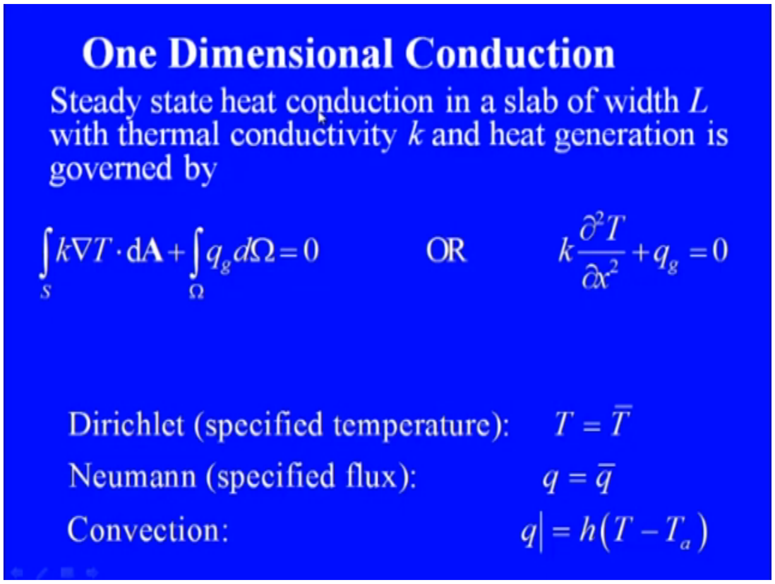
(Refer Slide Time: 05:53)



Now if we collect all the equations all discrete algebraic equations for all control volumes after incorporating the effective boundary conditions, we would get a system of algebraic equation form capital A times capital phi=Q, where capital A would be our coefficient matrix or system matrix, phi is an vector of unknown nodal values of finite computational nodes, and Q is known vector.

Now this vector A once again if you look back an equation for one particular control volume these are contribution coming from only few nodes which are the neighbor nodes of the computational node, so this matrix A is always sparse in fact is this bursty pattern is fairly similar to that obtained in finite difference method if we have used the structured grid, and would be fairly similar will be a banded structure if we have used an unstructured grid finite volume formulation.

(Refer Slide Time: 07:13)



Now let us come to an example problem which we will discuss in detail that how do we apply finite volume application method to solve a particular problem, so as a template let us say we take one dimensional heat conduction problem for the sake of simplicity we are going to restrict ourselves to steady state problem as for the basic discretization technique is concerned it would be exactly the same.

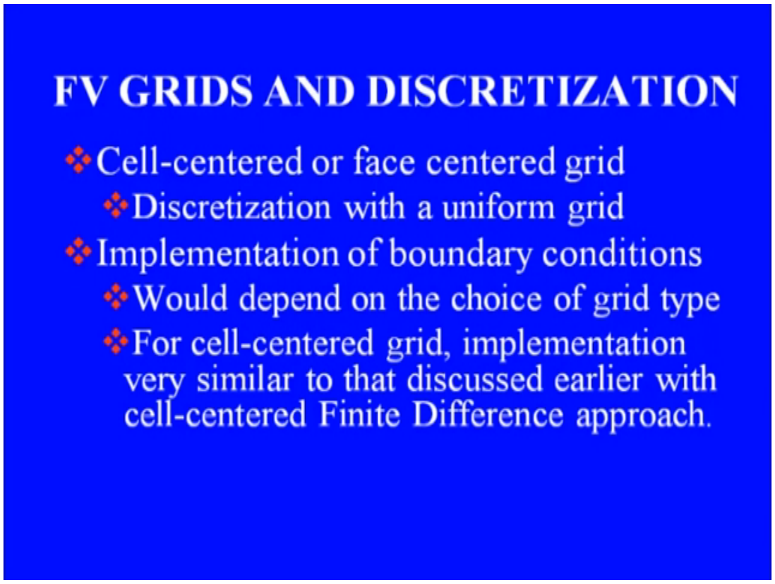
Even for a time dependent problem the only difference would be in the case of time dependent problems our discrete finite volume system would be a system of ODEs in time which we can solve using the time marching scheme, which we had discussed in the previous module okay. And let us assume that we have got a slab of width L with thermal conductivity k, and the heat generation qg.

If you are looking at the integral form for energy equation that is how it looks like integral over the surface k gradient of T dot dA+qg d omega is the volume integral which gives us the effect of the heat generation in the volume=0 or if this is a partial differential equation which we are get k del 2 T/del x square +qg=0, wherein we assumed k to be constant. And to obtain a unique solution for this problem you have to apply the constraints in the form boundary conditions.

So constraints could be in the form of Dirichlet boundary conditions of what we call a specified temperature at the boundary T=T bar, we can have at some part of the boundary what we called

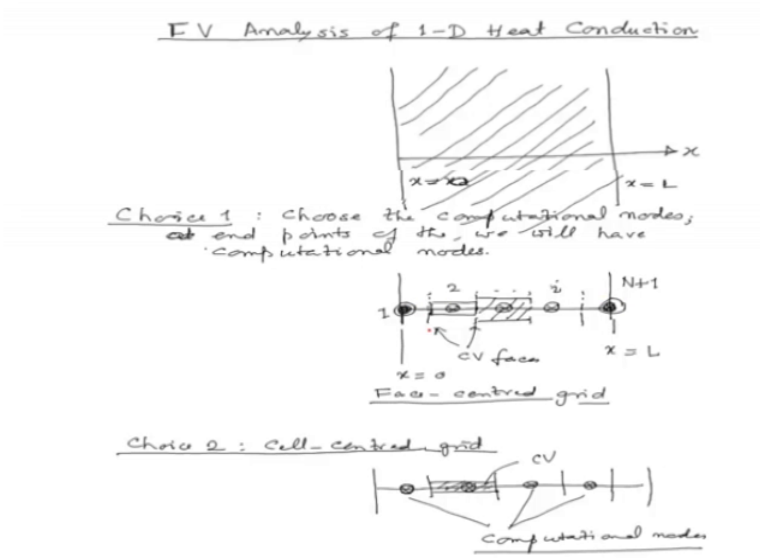
Neumann boundary condition or specified flux q=q bar, or we can have convective boundary condition specified in the boundary q at the boundary given in terms of the convective heat transfer coefficient h*T-Ta.

(Refer Slide Time: 09:09)



Now let us have a look at what choices we can make for discretization and how we proceed with finite volume formulation to solve this problem.

(Refer Slide Time: 09:19)



So let us have a look at on board this finite volume analysis of 1-D heat conduction, so this was our slab and we will have some boundary condition specified at both ends for timing let us not put any constraints okay. Now we have got two choices for the finite volume grids, choice one choose the computational nodes and at end points of the domain we will have computational nodes.

This is a way similar to the choice which we had made with our standard finite-difference formulation that this was our domain depending on the number of divisions we will have, we have chosen our computational nodes, and our control volumes would be taken to be the this mid-surfaces okay, so these are control volume faces, so for instance this becomes one control volume similarly, this one would become another control volume and so on.

So this is one choice and remember here at both ends let us call this these ends are x=0 and x=L, so this would we have got one node at x=0 let us call this node as 1, 2 and so on, so ith node p and this is the last node, so if you got N divisions the last node would be numbered as N+1 which corresponds to x=L, so this is one particular grid what we call this is our face-centered grid. We can make yet another choice cell-centered grid.

Now remember as we have discussed earlier the cell-centered grid is used more commonly in finite volume analysis, so here we have got the domain we will first divided it into cells, first let us from our control volumes, and the center of each control volume that becomes our computational node. So we first choice our CVs and then so this hatched portion was our CV, and these are our computational nodes.

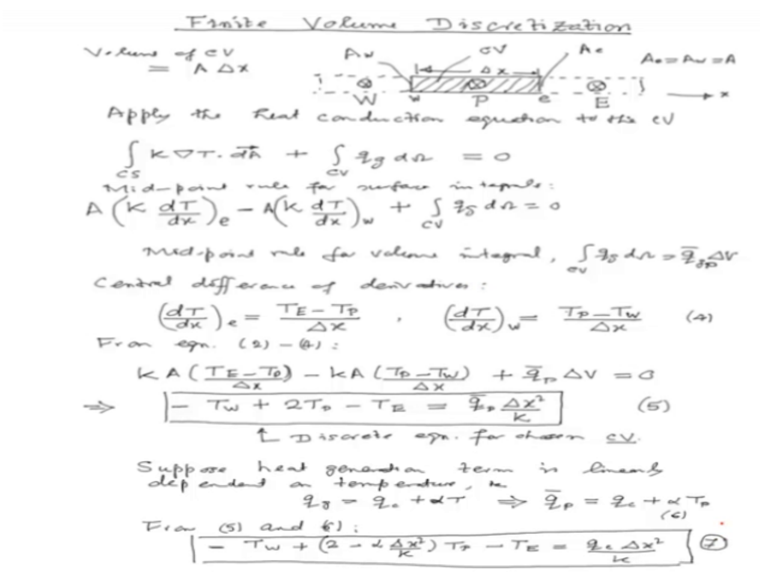
Now let us if you look at 2 things clearly as far as the basic discretization procedure is concerned our basic finite volume formulation is concerned that would remain the same, we will take a particular control volume, and we would apply our governing integral equations to obtain a discrete equation for that control volume. So whether we take a face-centered grid or a cell-centered grid in both the case the basic formulation would remain the same.

The only difference would be how do we create the boundary conditions? The treatment of the boundary conditions would be slightly different in the face-centered grid compared to the face centered grid. For instance we had Dirichlet boundary conditions specified both the ends of the slab in this face-centered grid where the vertices are coincident with the boundary ends, we get a very simple equation T1=Ta that becomes our discrete equation at let us say the node 1, and similarly, TN+1=TB okay.

And for imposing the flux boundary condition we will have to choose a slightly different computational volume close to the boundary, these aspects we are going to discuss next, but first let us have a look at the finite volume formulation for a typical control volume. So we will choose a discretization with a uniform grid okay, and we will implement our boundary condition this implementation would depend on the choice of the grid.

And as we will say that for cell-centered grid implementation is very similar to that discussed earlier with cell-centered finite difference approach. So now let us get a finite volume this grid equations for a generic computational node or we can say generic finite volume.

(Refer Slide Time: 16:26)



So let us take a generic finite volume its east face, and its west face, the centroid and this is the computational node let us call it P, if you have taken uniform grid with either option this face-centered option or cell-centered option the conditional node would be at the center of the finite volume, let us also draw a dotted line let us denote the neighboring CVs in fact we are interested in the neighboring computational node, this competition node capital W this computational node capital E.

And we would now apply our conservation law that is in this case our integral form of the heat conduction equation to our chosen control volume, so apply heat conduction equation to this control volume CV the area of the face of let us say this particular west face this area is Aw, the area of east face is Ae. And in this case note that Ae=Aw same, and what is the volume? Volume of CV this would be basically A times delta x where delta x is the extent of this control volume in x direction okay.

So first term was a surface integral K gradient of T dot dA+ we had a volume integral take care of qg d omega=0, now in one dimensional case now we have got only 2 control surfaces, so this K dot dA that becomes very simple, so this would become A*K times dT/dx at eastern face-A*K times dT/dx at the western face+ let us use our surface integral over CV qg d omega=0, and of course these are area integrals multiplied with the area A.

So here what we have done is we have used our mid-point rule for surface integrals, we can use a same rule to volume integral, so if we use the mid-point rule for volume integral then we will get this qg d omega CV=qg bar delta v where qg bar is the value of the qg at the centroid that is at point P, so you can also put the subscript capital P if you want to be very particular. Now we have got these two derivatives in the diffusion term dT/dx at and dT/dx at w.

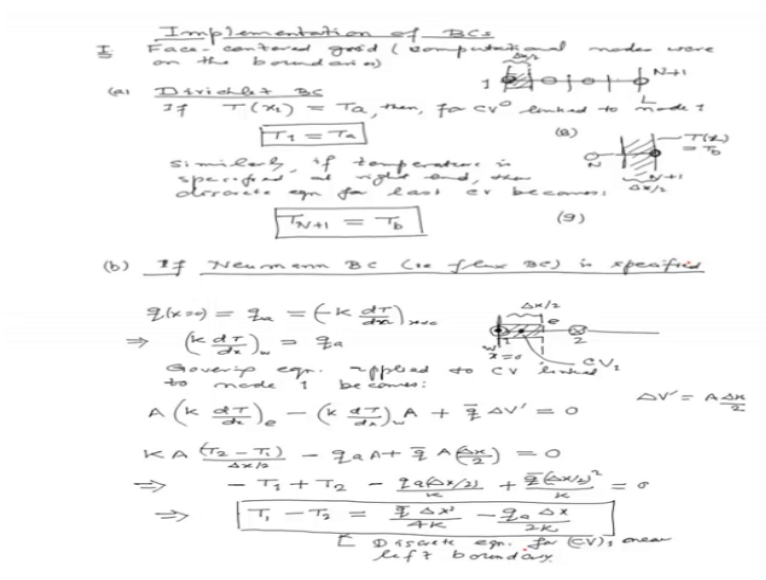
So let us use central difference approximation, so central difference approximation of derivatives, so dT/dx at e this is our east face it can be written in simply in terms of the value of temperature at eastern nodes e so that becomes TE- value at current computational node TP/delta x and similarly, the value of derivative at the west face because west face that represents the mid-point between the computational node P and W, so this is given by TP-TW/delta x okay.

So now let us substitute these approximations in equation 2, so from equations 2 to 4 what do we get? K we have assumed to be constant let us say K ATE-TP/delta x -K ATP-TW/delta x +q bar P delta v=0, let us simplify this expression we can divided by K A and multiplied by delta x, and rearrange the terms, so we will get –TW+2TP-TE=qp bar delta x square/K, so this is the discrete equation for our chosen interior control volume, so let us call this final equation as our equation 5.

This equation has to be modified a bit let us say first let us account for one particular case where we have the source term is temperature dependent, so suppose at heat generation term is linearly dependent on temperature that is we can write this as this qg=qc+ alpha times T, where alpha is appropriate proportionality constant. So now in this case our average value this qp bar would be nothing but qc+ alpha times TP okay.

So we can substitute it now in equation 5 let us call this equation as 6, so from 5 and 6 if you substitute it there and then collect the TP terms bring it on the left hand side we get –TW+ 2-alpha delta x square/K times TP-TE=qc delta x square/K, so this is the final form of our final discrete equation for a finite volume. And now this equation has to be modified by accounting for the boundary conditions for boundary nodes okay, so that would be the only difference and that if the implementation would depend on the choice of our finite volume grid.

(Refer Slide Time: 27:06)



So this is our implementation of boundary conditions and we will consider the 2 cases separately, the first one was this face-centered grid wherein this computational nodes where on the boundaries that is this was our domain and domain boundaries 0 and L, so the first computational node was on the boundary. Similarly, if we had N divisions so last computation node which with numbered as N+1 this also on the boundary.

Now how would we implement different types of boundary conditions, so let us take the case of Dirichlet BC that its temperature is specified, so if Tx1=Ta, then what happens for the different equations for our finite volumes, this one control volume which is linked to this boundary node 1 for that we have got a very simple equation, so then for CV linked to node 1, so this was our CV linked to node 1, we have got a very simple expressions that is T1=Ta let us call this equation as 8 an continuation.

Similarly, we had a boundary condition specified at the last and the right end, which was our node number N+1 and this node number N and this was our CV boundary, so CV of finite volume linked to the node N+1 that would have a very simple form. Similarly, If temperature is specified at right end then discrete equation for last CV becomes TN+1=Tb, suppose you have a specified temperature here is T L =Tb okay.

So in this scenario these are the equations which are to be put in for the control volumes and note down that these control volumes are half-width delta x/2, so for these control volumes which are adjacent to a boundary computational nodes. In the case of Dirichlet BC we have got very simple expression and these 2 equations can be put together with the discrete equation for other control volumes to obtain our final system and solve it.

Now instead of temperature specified suppose flux was specified, so if Neumann BC that is this flux BC is specified, then how do we incorporate it? In that case the treatment would be slightly different, so now let us have a look at our, suppose we are dealing with the left end that is x=0 and here the value of the flux is specified q x=0 suppose let us call it as qa it is given. So we have got this computational node 1, computational node 2 and so on.

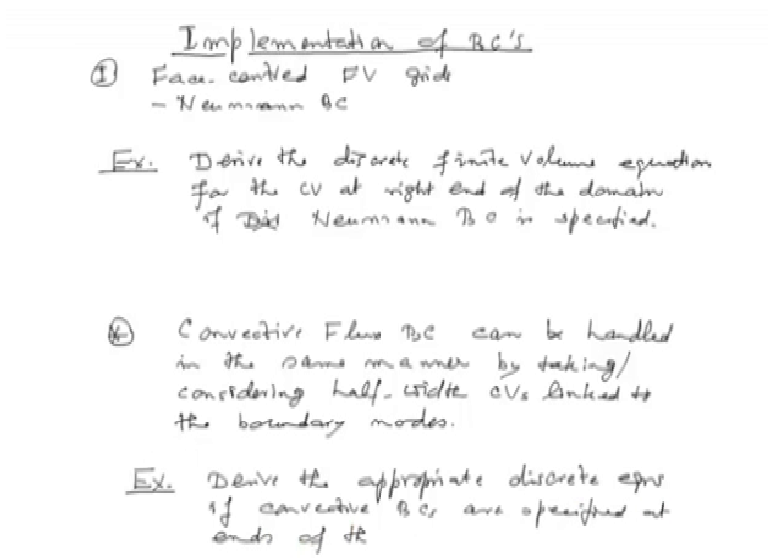
And now this becomes of a CV face and this is the half-width control volume which is linked to computational node 1, so this is our CV 1 and its width is delta x/2. So now we have to write a initio equations or we would apply our integral equation to this half-width CV and we would note down that the flux is given that is this qa=-k dT/dn at x=0 if you take care of the directions.

So this becomes our K times dT/dx for the west face thus, our boundary now coincides with the west face, this east face of the CV, so this is given as qa. So the equation for governing equation applied to CV linked to node 1 becomes A K times dT/dx at face e-A K times dT/dx at face w of course we have we will have to multiply that areas +q bar, where now this q bar is the average value of the value at the centroid of this control volume*delta v prime=0.

And we have to be careful that what is delta v prime now this delta v prime would be A times delta x/2, so please be careful about this and in the place of K times dT/dx we would substitute our qa, so our equation simply becomes K A now dT/dx let us use the central difference scheme in terms of the values at node 1 and the computational node 2, so we get T2-T1 and please be aware this is a half-width control volumes, so this is delta x/2-qa+q bar A delta x/2=0.

So let us simplify this expression further, so we get -T1+T2-qa delta x/2 times K+q bar delta x/2 whole square/K=0, on further rearrangement we can write it as T1-T2= q bar delta x square/4K-qa delta x/2K, so now this is the discrete equation for the first control volume, so discrete equation for CV 1 near left boundary. You can apply exactly the same process or same procedure to implement if the flux boundary conditions were specified at the right end, so I would leave that as an exercise to you.

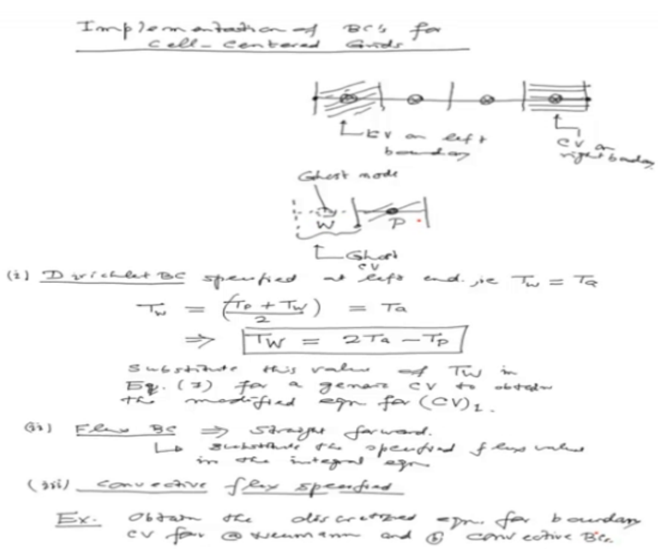
(Refer Slide Time: 38:20)



So implementation of boundary conditions remember we are dealing with this first case of face-centered finite volume grids and we had this Neumann boundary conditions. So your exercise is derived the discrete finite volume equation for the CV at right end of the domain if Neumann BC is specified. Convective boundary condition in this case can be handled the similar fashion, so that would leave it with just a comment there.

The convective flux boundary condition can be handled in the same manner by taking or by considering half-width CVs linked to the boundary nodes, and I would leave that as yet another exercise for you. Derive the appropriate discrete equations if convective BCs are specified at ends of the slab. Next, handling of the cell-centered finite volume is bit different from what we had seen with the vertex centered or face-centered finite volume grid.

(Refer Slide Time: 41:37)



So the implementation of BCs for cell-centered grids, in fact this concept we have already seen earlier when we dealt with the cell-centered finite difference formulation, so I would just repeat the basic approach which we use for implementation of boundary conditions in this case, so this would happen this we first divide our problem domain in the control volumes and their centroids now they become our computational node this is our cell-centered grid.

Now if boundary conditions is specified at the ends of the domain how do we account for this? Now these control volume exclusively linked to the points in the boundary, in fact we have 2 boundary CVs this is our left boundary CV or CV on left boundary and similarly, let us say this one becomes our CV on right boundary. So in this case what we will do is we will introduce the concept of a ghost cell okay.

Now this ghost cell can be taken on either side depending on which boundary control volume we have taken this is what we had discussed, suppose we take the CV towards the left end so this is our boundary this is our CV and this is computational node P, so we will take a ghost CV and its centroid becomes what we call let us say in this case this capital W this becomes our ghost CV and this becomes our ghost node.

And we would now apply the differencing or linear interpolation making use of the values at these ghost points, this is what we had done earlier in cell-centered finite difference implementation, this is the revision of what we had done earlier. So suppose we had Dirichlet BCs specified at left end, so that is what we have given is T at w that is given as the boundary conditions Ta, so how do we incorporate it?

Now what we can do is we can make use the concept of linear interpolation or averaging or we can say this now T at the west face this is the west face of this control volume, Tw is an average of TP+TW/2, so take the values divided by 2, so we get their averages, now this is what becomes Ta. And now use this from here we can get an expression for TW that is the value of the temperature at the ghost node this becomes 2 times Ta-TP okay.

And substitute this value of capital TW in equation 7 which we have derived for an interior node for rather we have derived for a generic CV to obtain the modified equation for control volume 1, we can adapt exactly the same process for the rightmost CV if the temperature were specified, so this were the first case if flux were specified flux BC, then of course we do not have to do anything we just go back to the beginning and keyword the face that is already known.

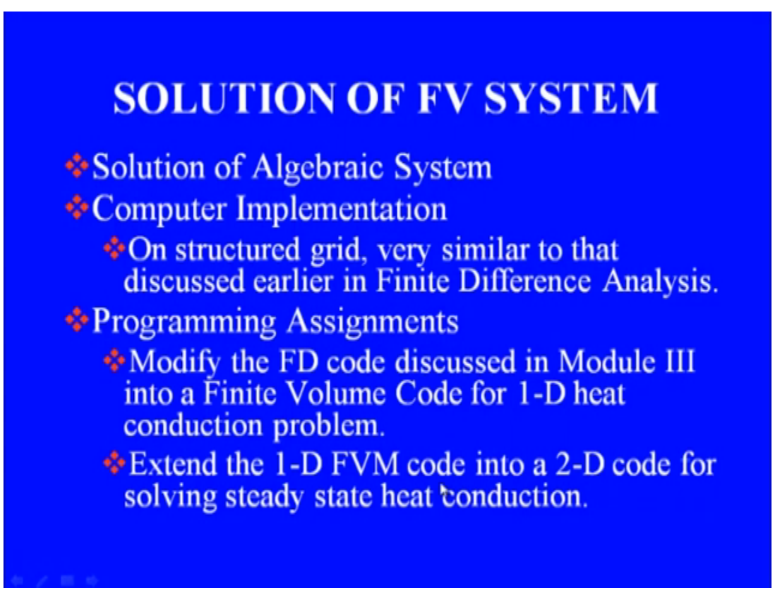
Substitute that value in application of the governing equations for this particular control value and we will get the modified equation for the boundary control volume for, so the flux BC is straight forward substitute this value this specified flux value in the integral equation, so it is exactly the same which we have the way we handle it for our face-centered grid it is exactly the same process nothing to worry about.

If we had the convective BCs for convective flux is specified okay once again the process remains similar to what we had done with the flux BC go back to our integral equation substitute that value, and there would be small catch which we will get there we have the value or at the end point of the domain that is what is involved that face temperature would be involved, and now the face temperature can be obtained using this straight line everything procedure which we had used for our Dirichlet boundary conditions.

So I would leave these expressions as an exercise for you, so obtain the discretized equations for boundary CV for 'a' Neumann and 'b' convective BCs. I would just again like to remind you that these expressions will be fairly similar in the form to what we had already derived earlier in detail when we were dealing with cell-centered finite difference formulation, so that is the reason why? I have given this as exercise to you.

So now we are done with the derivations of our implementation of a boundary conditions and once we are done with collect all the equations in this case you can easily observed that each equation contained the values of two neighboring computational node.

(Refer Slide Time: 50:18)

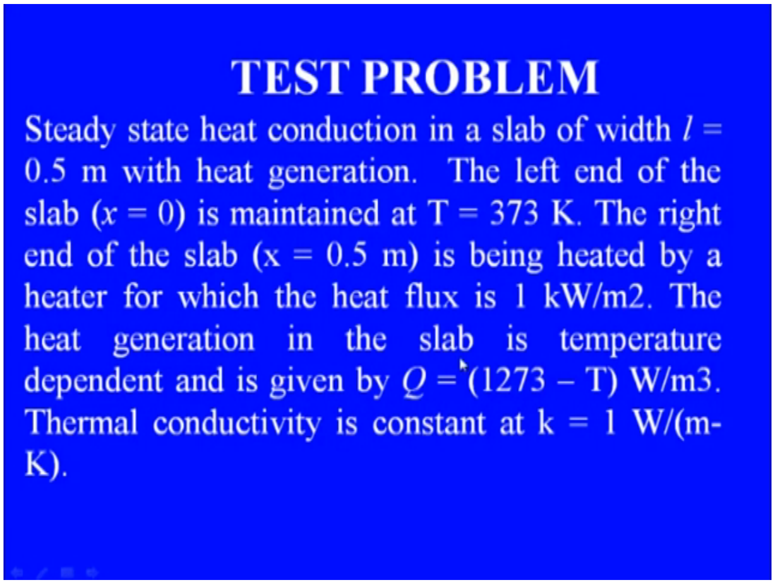


So we will get what we call a tridiagonal system we have already learnt TVM algorithm, we have also seen the program based on it, so you can easily use that routine to solve it, now as for this computer implementation is concerned is almost identical to what we had done in the case of finite differences, so structured grid finite volume or structural grid finite difference in both the cases the implementation of the computer program remains almost identical.

So we are not going to have a detailed look at it because you have already seen a detailed design of the code okay, so please note this similarity and based on this I would like to give you assignments that we have already discussed in detail how do we design a finite difference code the structured grid finite volume code is very similar to modified that code wherever the modifications required the FD code which we discuss in module 3.

And turn it into a finite volume code for 1-D heat conduction problem, we are also given development of a structured finite difference for 2-D problem, so extend that or extend your 1-D finite volume code into a 2-D code for solving steady heat conduction problems. In the end let us have a look at the numerical results for a test problem which we had already solved earlier using finite differences.

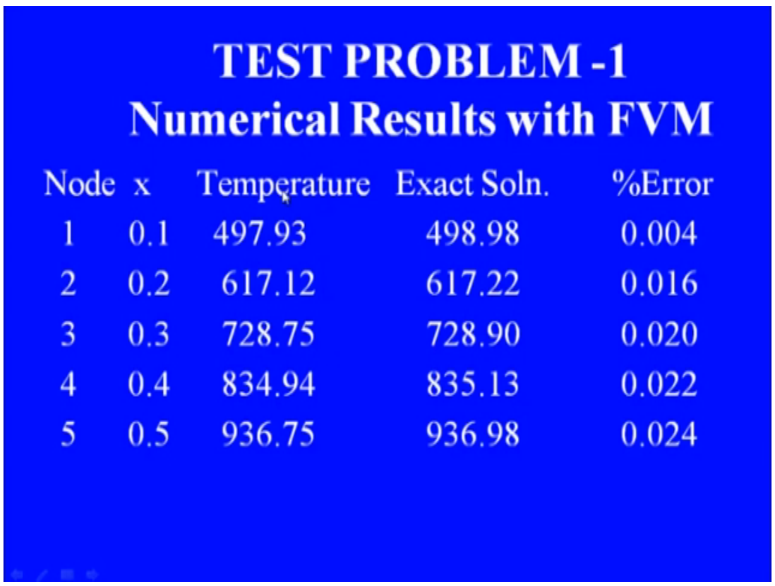
(Refer Slide Time: 51:40)



This was the slab of width l=0.5 meter with heat generation. The left end we had this Dirichlet boundary condition specified that is it was maintained at T=373 Kelvin. The right end of this slab was heated by a heater with a flux given as 1 Kilowatt per meter square. And the heat generation in the slap is temperature dependent capital Q this is qg basically = 1273-capital T watt per meter cube. Conductivity is constant.

So this same problem we have solved using our finite difference schemes earlier, and the finite volume formulation which we have just discussed if you implement it this is what you should get the numerical results with finite volume method at interior nodes, this is based on the face-centered formulation where in the nodes are the computational nodes coincide with the boundaries as well.

(Refer Slide Time: 52:47)



TEST PROBLEM -1
Numerical Results with FVM

Node	x	Temperature	Exact Soln.	%Error
1	0.1	497.93	498.98	0.004
2	0.2	617.12	617.22	0.016
3	0.3	728.75	728.90	0.020
4	0.4	834.94	835.13	0.022
5	0.5	936.75	936.98	0.024

So we have got the values put here at these interior nodes x=0.1, 0.2, 0.3, 0.4 and the last node 0.5. And these temperature values are in this particular column they are the values obtained from finite volume formulation this for the exact solution. And errors you can just compare these errors we have already got the relevant slides with finite differences, you can go back and compare.

And you can easily say that the results obtained with finite volume formulation or more accurate compared to finite difference formulations, and the reason was very simple in this case in finite volume formulation we have used second order accurate schemes everywhere, whereas in the case of finite differences there was at the boundaries for the flux term we had used a backward difference approximation which was only first order accurate.

(Refer Slide Time: 53:47)

So we will stop here, with our discussions on finite volume method. And these are our references, if you want to have for the detailed study of finite volume method please refer to this books Computational Fluid Dynamics by Chung, or book by Ferziger and Peric and book by Versteeg and Malalasekara. Specifically, in this introductory course we would not be discussing finite volume formulation on unstructured grid which are relevant to complex geometries, so for finite volume method applied to unstructured grid please refer to any one of these books.