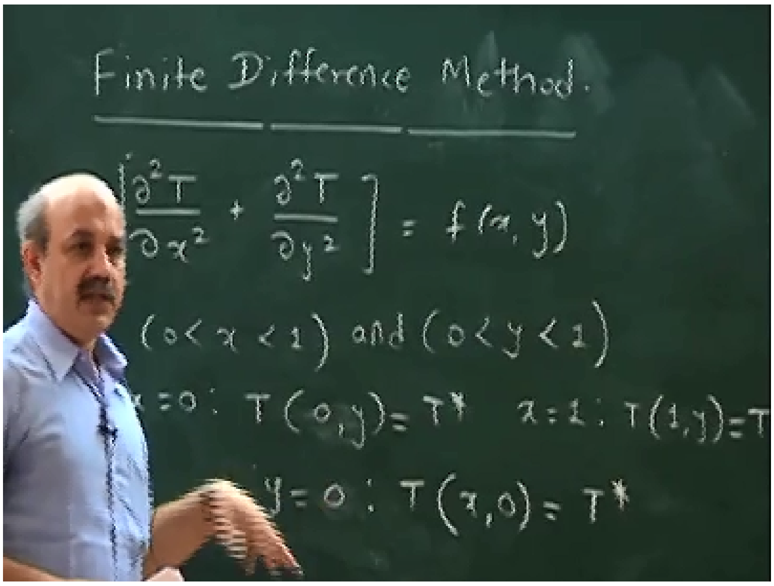
Advanced Numerical Analysis
Prof. Sachin Patwardhan
Department of Chemical Engineering
Indian Institute of Technology – Bombay

Lecture - 14
Finite Difference Method (Contd.) and Polynomial Difference Equation

So we have been looking at finite difference method that is used for discretizing boundary value problems as well as partial differential equations. And in particular in my last lecture I was looking at a partial differential equation Laplace's equation and then we approximated the partial derivatives in spatial direction and then what results is the set of linear algebraic equations.

So we start with a partial differential equation and then process of discretization resulted in set of linear algebraic equations.

(Refer Slide Time: 01:06)



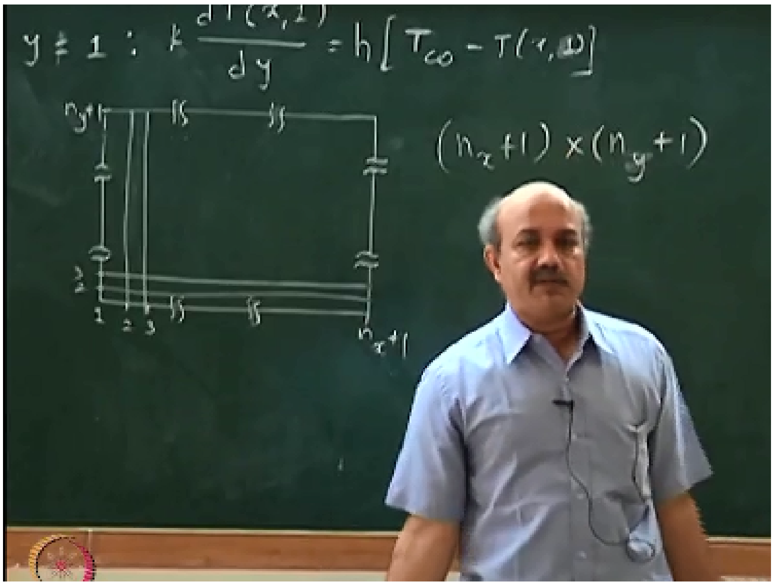
The problem was so we had this partial differential equation so x and y are normalized spatial coordinates and then we have this partial differential equation that holds on the interior points and then I said this was example of a furnace and then we use finite difference method to discretize this. What you get here is a set of linear algebraic equations. What I want to show again with the same example is that it depends upon the method you chose to discretize.

It is not that this particular problem and infinite difference method will always yield set of linear algebraic equations I am going to do one more method called method of lines which will yield ordinary differential equation boundary value problem. So depending upon how you choose to discretize you will get different types of approximations from the same problem that is very important.

And using the same method, same approach that is finite difference of course when we use some other method which we will be discussing for example starting from towards the end of this lecture and next lecture. We will talk about orthogonal collocation so that is another way of discretizing that of course will yield different way of formulating the approximate problem.

So here we had these boundary conditions at x=0 we have condition that T 0y =T* then x=1 we have this condition T 1y=T*. So it is like saying that 3 walls are insulated then there is convective heat transfer from one end at y=0 we have T temperature at x0=T * so 3 walls are insulated.

(Refer Slide Time: 03:35)

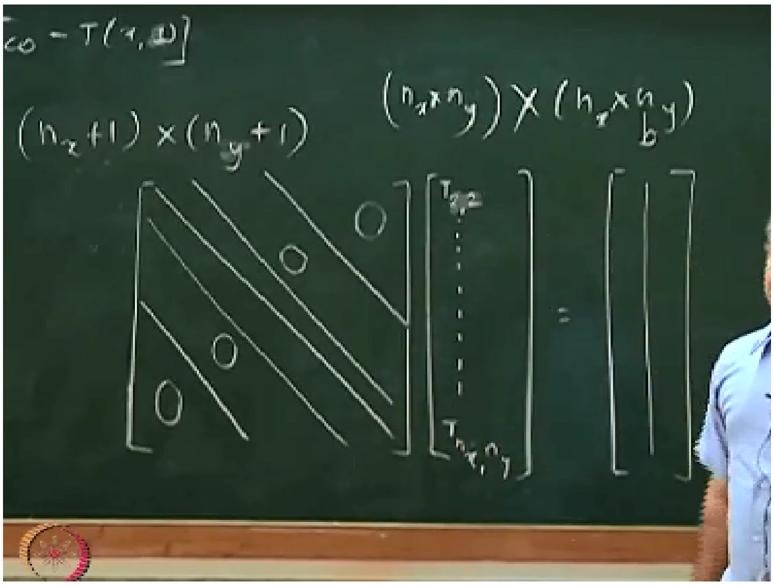


And the fourth one we have convective at y=1 we have this 4 boundary conditions. So we have these 4 boundary conditions and when we use finite difference method what we got was set of linear algebraic equations. What we had done was we discretize the domain. So this is 1, 2, 3 and so on. So we discretize the domain we constructed, we demarcated grid points along y axis and along x axis.

And then we impose the partial differential equation at the grid points or in the terms of finite difference method. We force the residual to 0 at the grid points and this forcing residual to 0 at the grid points gave rise to large number of algebraic equations and same thing is to about boundary conditions we used finite difference at the boundary and then we get additional equations at the boundary.

So total number of equations that we have finally is how many equations you have how many variables. You have nx+1 cross ny+1 equation and you have same number of variables. Now instead of writing those equations again I want to just point out something if you actually try to rearrange this what you get is set of linear algebraic equations and if you rearrange them into a matrix form the standard you know Ax=b form then you get a very special kind of matrix

(Refer Slide Time: 05:45)



I do not want to write the matrix. I just want to show how it looks graphically. It is a special diagonal matrix it is a sparse matrix what you get here and so this matrix will have non 0 elements along first 3 diagonals there will be lot of 0s there will be non 0 diagonal again there will be 0 here, 0 diagonals and 0 diagonals. So you get a sparse matrix here if you rearrange all the equation. So this is T 1 1 or I think you have to eliminate the boundaries.

So you get T 2 to to T nx ny. If you eliminate the boundary conditions and then collect all the unknowns together and write this as 1 equation you will get a vector b here which is coming out of F of xy at different points f of xy at different points. So here this in this case this is the source term and as I explain to you that the sources if you are modeling this particular room then the sources would be each one of you is a heat source like a 40-watt bulb.

And the distribution of these bulbs can be given by f xy you know at which points these bulbs are located and then that would lead to a particular solution, a particular temperature distribution. So the solution here actually the true solution is a surface with one temperature value attached to each point each xy in this. We are not able to solve this equation at every point.

We are able to solve it only at finite number of points. You will get a good solution if you create more numbers of grid points. The solution will improve with more and more number of grid points, but their computations increase with more and more number of grid points and this matrix which you see here which you will get will be a banded matrix and this will be a sparse matrix.

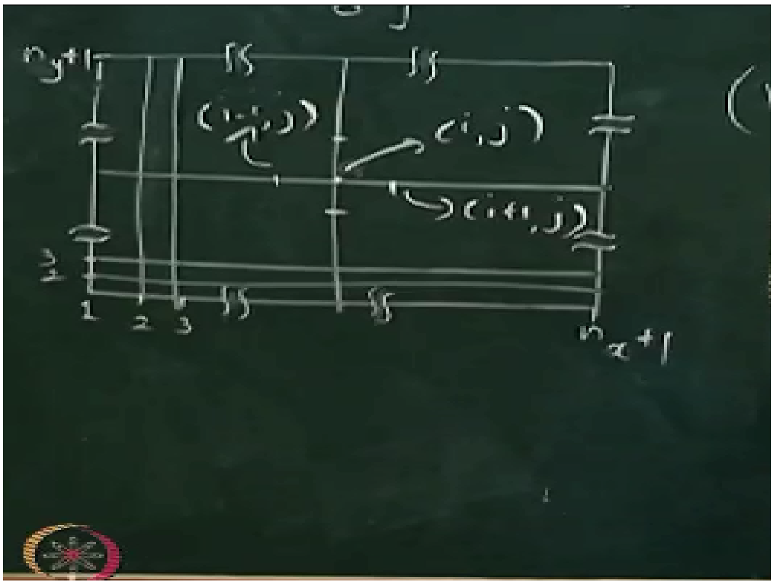
This is a side note that we will be looking at sparse matrices little later, but I just want to point out that I just want to provide a motivation for looking at sparse matrices, why do we have to look at special methods for matrices which are filled with large number of 0s because you can save computation time when you have large matrices. Suppose this nx and ny this happens to be these are the number of variables.

And this matrix will be nx cross nx these are the number of variables. So this matrix will be nx cross ny cross nx cross ny. This will be a huge matrix just remember. Total number of variables are these. So this if you eliminate boundary conditions you will get something like nx cross ny variables in this vector and this matrix is nx cross ny cross nx cross ny. So if you take 100, 100 grid points just imagine what is the size of this matrix.

Inverting this kind of matrix is possible, but you are wasting lot of computer time because there are lot of 0s and those will probably give 0 when you invert. You do not have to waste your energy in finding a 0s on the right hand side. If you device some intelligent algorithm you can save computation time. Now this is the only way of discretizing. We discretize both in x and y.

We discretize both in x and y. First of all, why do I get this banded structure.

(Refer Slide Time: 10:04)



Because when I discretize each equation will have only 5 variables associated with it. So I will have this variable, this variable, this variable, this variable, this variable. So this is I this point is i, j this point will be i+1, j this will be i-1, j and so on. So 1, 2, 3, 4 points neighboring points are appearing in each equations and such you have large number of coupled equations you have to solve them together simultaneously to arrive at a solution. The solution surface is being approximated by solving the equation at discrete number of points.

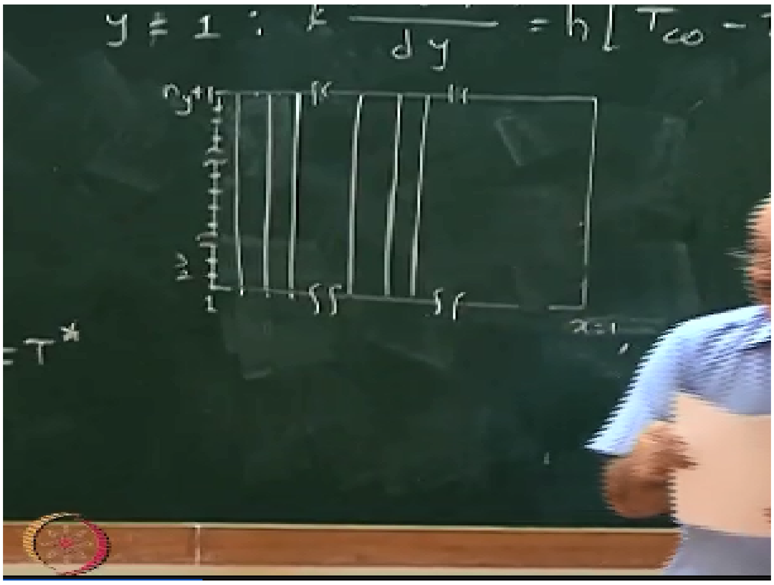
Since it is an approximation what you get is the approximate solution cannot be the true solution. For this particular problem in many cases you will be able to solve the problem analytically that is a different story, but if there is a non-linearity then you are in trouble for example I can make this problem non-linear just by saying that this thermal diffusivity alpha is a function of the function of temperature.

Right now I assume constant temperature probably if you are doing a modeling in this room the temperature variation is not too much it is a good assumption that alpha is constant, but if there is too much variation of temperature in a region where you are modeling then it is worth modeling alpha as a function of temperature and then immediately the equations which are linear algebraic equation will become non linear algebraic equations.

Okay leave that aside. Let us not talk about non-linearity. Let us look at this problem again through another method called method of lines. So what I am going to do now this is a method which belongs to the same finite difference class or it could reappear when you do orthogonal collocations instead of discretizing in both dimensions x and y. I could choose to discretize only in 1 dimensions say x.

And I want to retain the differential operator in Y direction. I will discretize only in x direction. I want to retain differential operator in y direction. This is another way of discretizing. So this will give me method of lines.

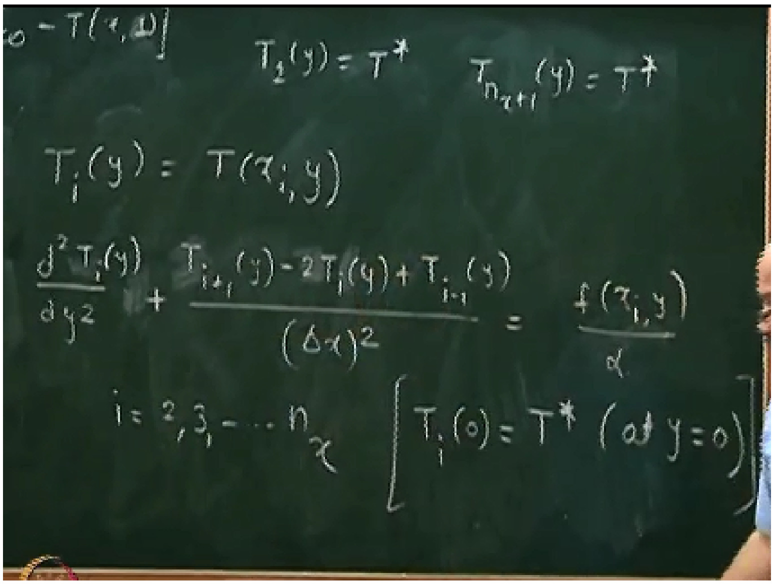
(Refer Slide Time: 12:36)



So what I am going to do here is instead of drawing grid points in both x and y direction I am not going to mark. I am going to draw parallel lines I am going to mark grid points along y and I am going to draw parallel lines here. Well the lines in my drawing are not looking parallel, but I intend to draw parallel lines. So I am going to draw parallel lines here. I want to discretize only in 1 direction and not in the other direction.

So I have made a mistake here. Well I want to draw lines parallel to x axis. So I want to draw lines parallel to x axis. Well it is not really important whether you discretize you keep you discretize in x direction or whether you will discretize in y direction it is not that important. In this particular problem because of a symmetric boundary conditions it is good to discretize along x direction.

(Refer Slide Time: 14:09)



So what I am going to do now. I am going to mark this variable now this is my notation. My notation is Ti y is a variation of temperature along y direction in along the ith line parallel to y axis. I have drawn lines parallel to y axis and Tiy is the variation of temperature along ith line parallel to the Y axis. I am just fixing x1 so this is my x=xi this is the ith point and Tiy is nothing. The way I want to discretize this equation now is now look at this equation the way I have discretize this I have discretized only in x spatial coordinate x.

I have chosen not to discretize in y. Now I have converted the partial differential operator along y to total differential because now x has been taken care because of discretization. So this becomes an ordinary differential equation. Now we have to use boundary conditions and because of boundary conditions you will get is this 1 ordinary differential equation how many you get at all the interval gird points.

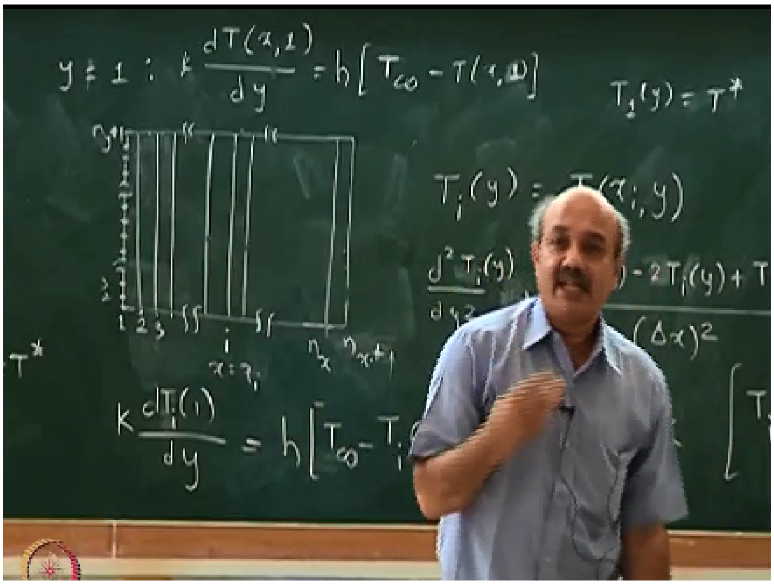
So 2, 3, up to nx and this will be the last point is nx+1. So till nx till last parallel line we get this differential equation. S I goes from 2, 3 up to nx. So you get ordinary differential equations not 1 ordinary differential equation you get coupled set of ordinary differential equations right. ith value or Ty Tiy is related to Ti+ 1y and Ti- 1y right. So the temperature on ith line is a function of temperature on this line and this line.

So what you get here is a set of coupled ordinary differential equations not 1 ordinary differential equation. Okay you get large number of coupled ordinary differential equations Suppose we discretize with 100 grid points internal grid points 100 coupled ordinary differential equations need to be solved simultaneously okay because neighboring variables appear in each one of them.

And now we can use the boundary condition to complete the problem. So we have this 2 boundary conditions T1 Y is= to T* and T nx +1 Y is= to T*. You can use these 2 conditions to eliminate some variables or you may have to solve it as a differential algebraic system okay. So you have these conditions in addition + you will have to have 2 conditions which will now become boundary conditions for his ordinary set of ordinary differential equations.

So those will arise because of this and this. So if I just write them in. Well we do not have space here okay I will write it here. This is the one of the boundary conditions.

(Refer Slide Time: 18:37)



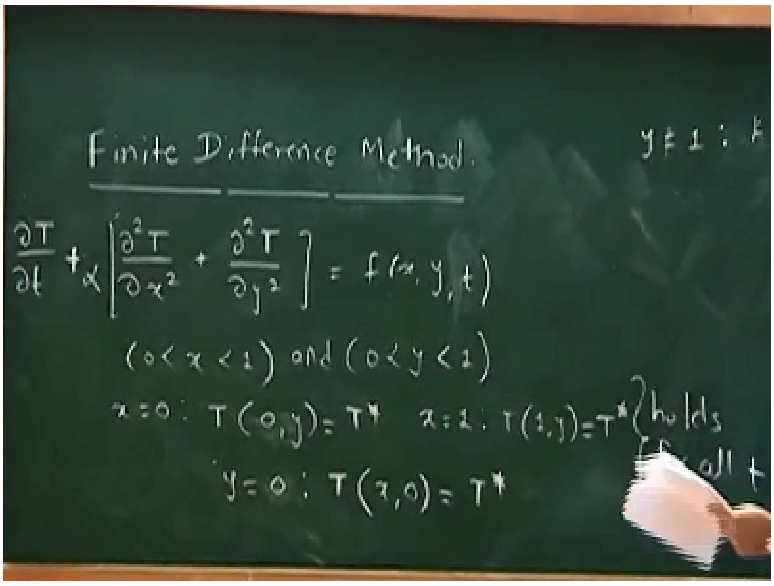
And the second one will come because of discretization of this and that would be K Ti1. So discretization of this boundary condition will give you this set of equations. Now what do you get here is a set of coupled second order ordinary differential equations which are these subject to boundary conditions at y=0 at y=1. So same problem finite difference method, but instead of choosing to discretize in both the direction if I discretize only in 1 direction I get OD boundary value problem.

Same original PD I get OD boundary value problem. So it depends upon how you choose to discretize. This problem could yield set of linear algebraic equations if you choose to describe in one particular way this problem will yield set of ordinary differential equation, boundary value problem. If you chose to discretize in another way. So it depends upon how you chose to discretize the problem.

Now I am going to write one more variation of the same problem, but we have been looking at the steady state equation what if I decide to look at in addition time variation. This is the steady state Laplace equation what about time variation if I include dou t well then there is a problem in using method of lines because if we discretize only in 1 dimension you will get another partial differential equation so which you again have to discretize.

So it depends upon the problem at hands.

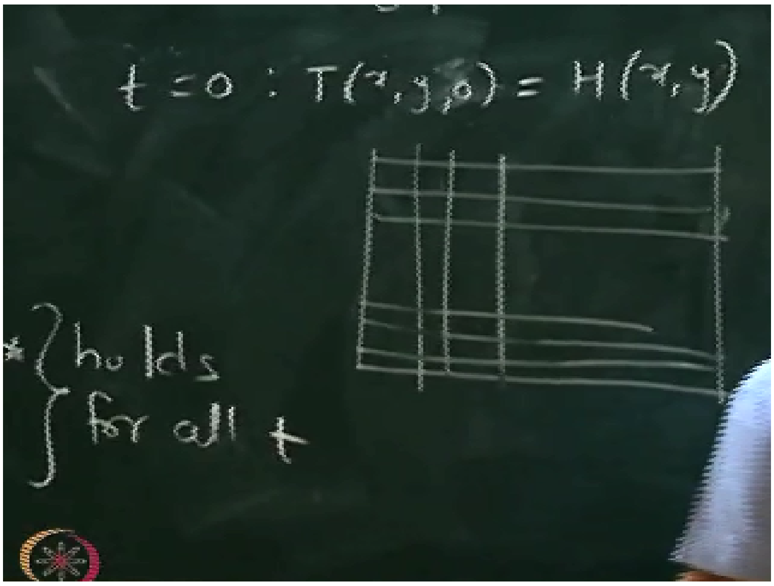
(Refer Slide Time: 20:45)



So suppose I have here a modified version of this problem. So the modified version I would just create here itself. If I add here and my right hand is instead of f of xy I will make it time dependent and all these conditions at boundaries will hold for all time. And then I will also have initial condition so I will have to give an initial distribution of temperature in the room. So what I want to know is how the temperature surface is changing as a function of time.

When I am including time derivative here I am looking at the spatial distribution of temperature in this room or in the furnace or whatever the condition is. And then I am interested in time evolution of this temperature not just the steady state. So this problem will also have additional initial conditions coming up. So the boundary conditions have to hold for all t. So this has to hold for all time t including this boundary condition.

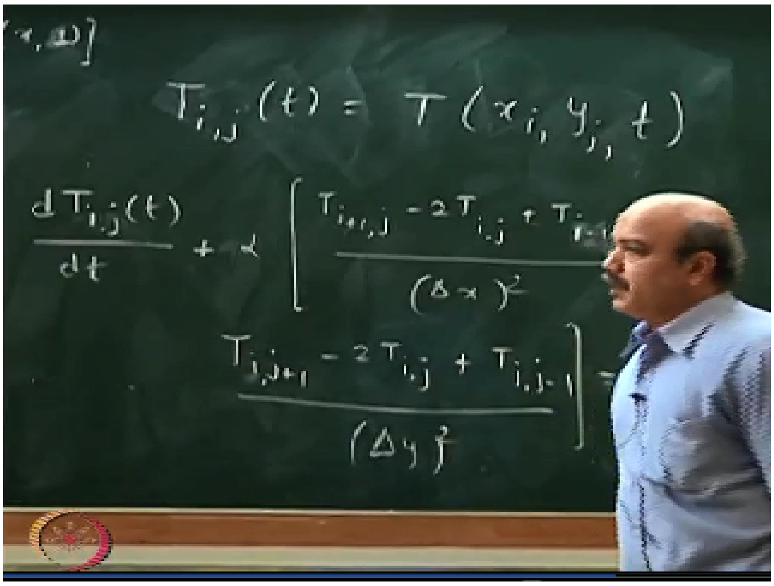
(Refer Slide Time: 22:19)



And then you will have at t=0 you will have temperature at xy0= some functions say H xy. So this is the initial condition. So this is the initial distribution of temperature in the room and then I want to solve this problem what should I do here can you suggest something? What if I decide to discretize in 2 spatial dimensions? I discretize in x and y. I leave temperature as it is what will I get? I will get ordinary differential equation initial value problem.

Large number of ordinary differential equation which are coupled. So here if I now take I am not going to completely write the solution you can work it out I am just going to hint at the solutions. So I am going to call now Tij xi or I am going to call Tij t=temperature at xi yj and t. Like in the previous case I have created a grid here and ij-th point the time variation I am going to write like this.

(Refer Slide Time: 24:05)



And then well I will discretize this problem as dTi, j t/dt+ alpha times. Now I will approximate as Ti+1, j-2 T ij+ Ti-1,j/delta x square+. So if I discretize in 2 dimensions I get large number of ordinary differential equations in time, but in this case I will not get a boundary value problem just think about it. In this case I will get a ordinary differential equations initial value problem.

So write now we are not talking about how to solve this problem. We are just talking about transformation. We want to take a problem transform it into a solvable or computable form. How to actually compute that solution will come to that later. I want to separate these 2 things. I just want to give you a viewpoint that well if I take a partial differential equation or ordinary differential equation I know from (()) (25:39) theorem that any continuous function can be approximated using polynomials.

So I used Taylor series approximation construct local approximation of the differential operator and force the so called residuals to 0 that give rise to an approximate problem. Approximate problem would look completely different from the original problem. A PD is giving rise to set of algebraic equations, linear algebraic equations. PD giving rise to set of ordinary differential equations boundary value problem or a PD giving rise to set of ordinary differential equation initial value problem.

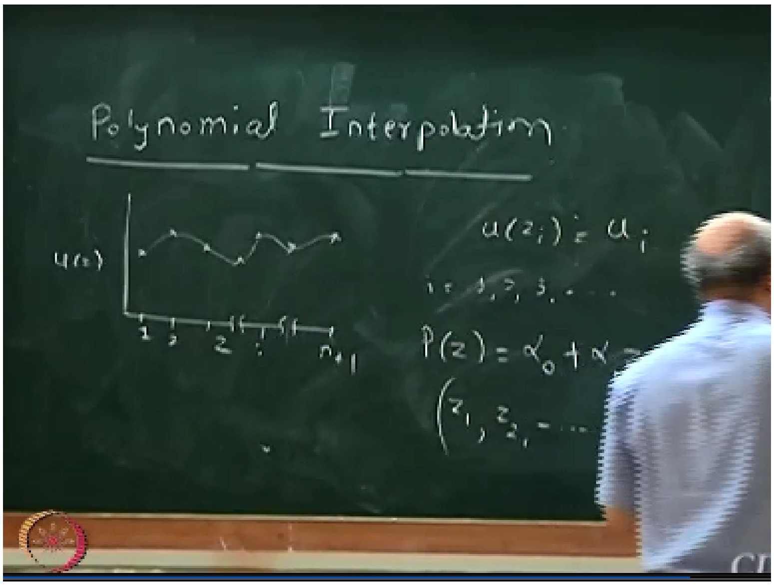
So we bring all these into some standard forms and then we apply the standard tools to solve this problem that is what you should realize. There is no unique way of coming up with a discretization. You could have your preferences solve the problem at hand. So in this place you can complete the initial conditions you can remove the so this called a DAE differential algebraic system with initial value problem, initial condition specified or if you eliminate.

It could be just ordinary differential equation initial value problem. So till now we are looking at Taylor series approximation and as I said this is only one of methods that are used for discretization other 2 methods are interpolations, polynomial interpolation and then least square approximations. So we will stop here with Taylor series approximation we will now move on to Polynomial interpolations.

And then see how interpolations can be used to discretize the same set of problems. What I have deliberately done is in the lecture notes is that I have taken same set of problems and then discretize them using different approaches so that you will get a better insight into what has really happened at the discretization stage. Now let us move on to interpolation. Interpolation is something which probably you are introduce in your undergraduate curriculum.

Can anyone tell me what is interpolation what is an interpolating polynomial. When do you interpolation or interpolating function you are given a set of points and values. So you are given some set of so you actually want to develop an approximation for a function continuous function. Let us take a continuous function say U or.

(Refer Slide Time: 28:23)



To be very precise let me see to begin with polynomial equation. Well you can also do function interpolation in general functional interpolation, but we are interested right now in polynomial interpolation because we use the polynomial interpolation ideas to come up with approximations or discretization of some operator. So now polynomial interpolation is we are given some values of say dependent variable Uz and independent variable z.

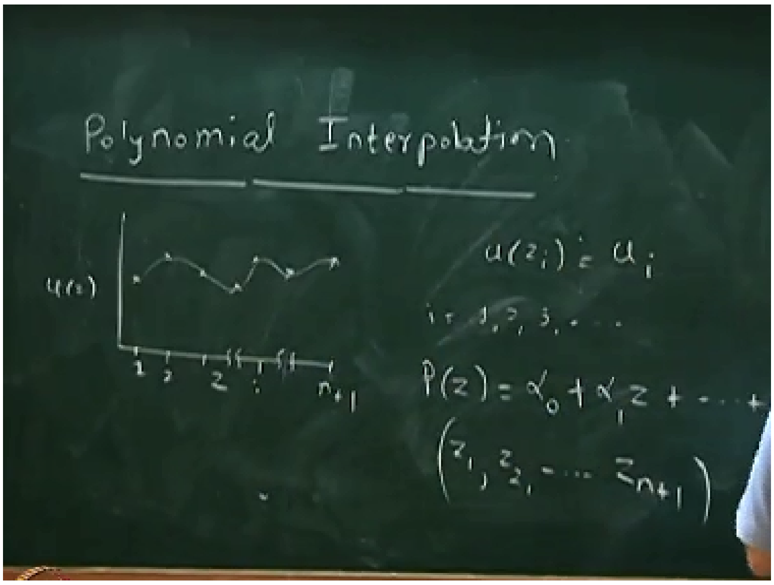
So we know some values of U just before I move on to this let me clarify one point in finite difference method I kept on using equispaced grid points. It is not necessary that you should have you should have equispaced grid points that was only for the convenience of development understanding actually you can use non-equispaced grid points. For example, now when to use non-equispaced grid points and then how to space them will depend upon your understanding of the physical problem.

So for example if you have a PFR if you have plug flow reactor and let say most of the action is in the initial part of the PFR and later you know the concentration becomes steady or it does not change too much it becomes flat. It is worth putting more grid in the initial part which are closer to each other and then putting sparse grid points later because there is not much thing happening at the end of the reactor.

So this depends upon your understanding of the problem. In some cases, you may want to place close grid points in the 2 ends sparse grid points in the middle it depends upon the problem. So there is sacrosanct about equispaced grid points. Equispaced grid points only make development on the board simple, but in general and as a beginner you might start with equispaced grid points, but then if you know about some problem.

Where there is more variation of a particular variation in particular region you can put closer grid points and then can have a sparse grid point that depends upon your understating of the system. So let us move on to this polynomial interpolation.

(Refer Slide Time: 31:00)



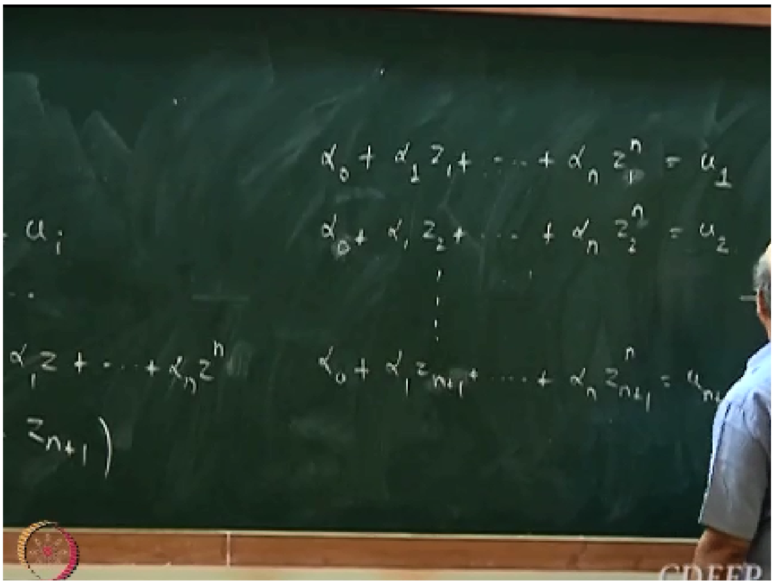
Now what I want to do here is I have this function uz as a function of independent variable z and I know the function values at these discrete points in the domain. So this is my point number 1, this is point number 2. This is in general ith point and this is my n+1 points from z 1 to zn. I know value of a function what I want to do is I want to construct a polynomial which passes through each one of them.

I know this is a continuous function of independent variable z. My theorem I can approximate it as a polynomial function. I want to construct a polynomial function that passes through all the points this is different from least square approximation which you do in your experimental work where you fit align. So the line may not pass through every point that is least square approximation. We will look at least square approximation little later.

But right now we are concerned about constructing a polynomial that goes through every point. So my problem is to construct a polynomial Pz I am going to construct a polynomial approximation Pz. Now this is alpha 0, alpha 1 if I have n+1 points I can fit a polynomial of order n that exactly passes through all these points. I need a polynomial of order n because the polynomial must pass through every point so this is a constraint.

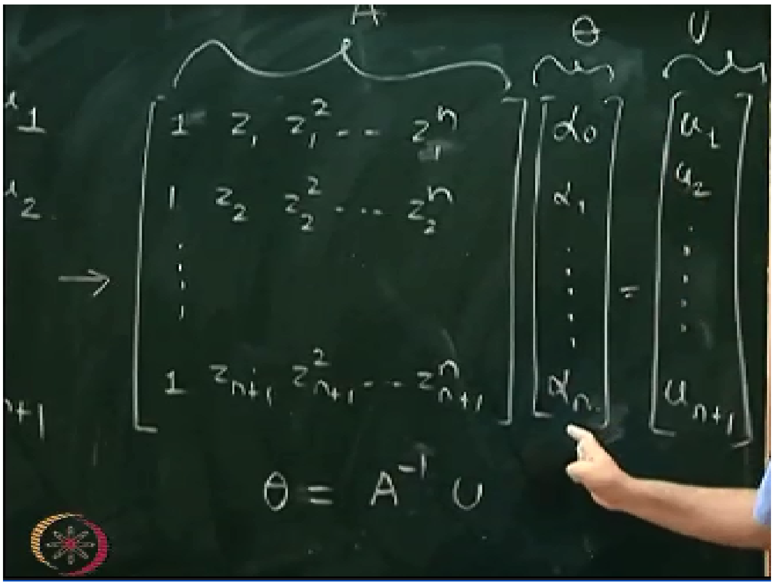
So what does this constrain means. So I have these points z1, z2, zn+1 I have these points what I want is that value of this polynomial at each of these points should be exactly equal to ui that is an interpolating polynomial. So I have this state of constraint.

(Refer Slide Time: 33:45)



So alpha 0+ alpha 1 z1 up to alpha n z1 raise to n=u1. Alpha0+alpha 1 z2 and so on. So I get n+1 equations I get this n+1 equation in n+1 unknowns what are the unknowns here alpha 0, alpha 1, alpha 2 up to alpha n. There are n+1 unknown and there are n+1 equations how do you solve this.

(Refer Slide Time: 34:30)



This can be simply transformed into 1 z1, z1 square z1 raise to n 1 z2 z2 square z 2 raise to n. So this is simply solving Ax=b. Well unfortunately life is so simple because it turns out that if you want to develop a higher order interpolation polynomial nth order interpolation polynomial in general about 3 or 4 this matrix here it looks very simple right Ax=b. This is known to you these values are known to you function values are known to you right now.

What you do not know is alpha 0 to alpha n and you know this matrix right you know the point at which so you know z1, you know z2 so you can create this matrix. We will see in the course of looking at ill-conditioned matrices. We will see that this is one of the highly ill-conditioned matrix.

Difficult to invert you can get so high order interpolation polynomials if you just go by this brute-force approach can lead to problems. Well one way to get out of the situation is to use orthogonal polynomials these polynomials are not orthogonal so that is why you get into trouble. If you use orthogonal polynomial you can come out of this problem and then this matrix is very nicely behaved and so on.

But right now let us not worry about that part right now let us assume that this matrix is invertible. So how do I get alpha 0 to alpha n so if this is my A matrix and let say this is my capital U vector and if I call this simply as theta parameter vector then at least on paper I can write theta= A inverse U. Where A matrix is known to me U vector I know the function values at these points.

Theta I can get what I get is the interpolating polynomial. Well I am going to use the interpolating polynomial to discretize boundary value problems, to discretize partial differential equations what I will be getting there or what the method that is used to do that is called as method of orthogonal collocation why this word orthogonal comes into picture. We will look at those details.

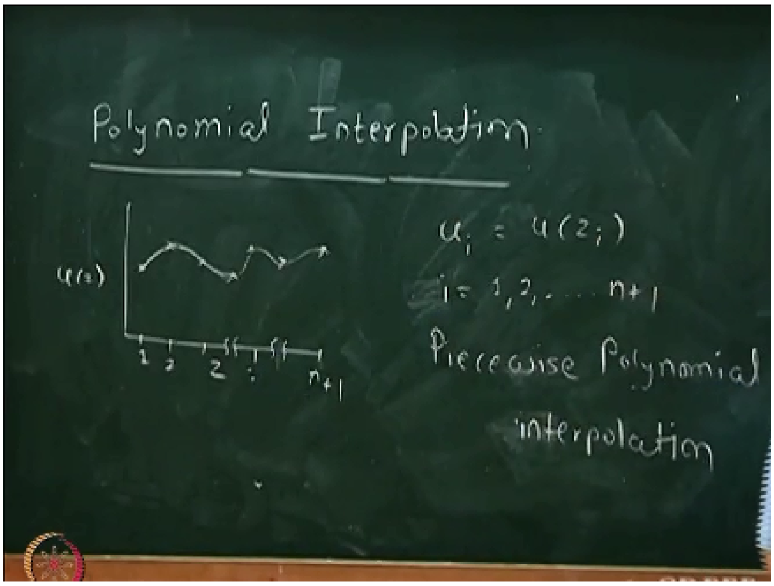
But the basis is this that if you know a value at certain number of points then theta can be expressed as A inverse U and this polynomial which you get here if you are able to calculate all these alpha 0 to alpha n correctly this polynomial will pass exactly through all these points that is very important. This polynomial will pass through each one of these points. It is a nth order polynomial that passes through each one of these points.

Now the trouble I told you is that constructing a very high order polynomial through large number of points is can you give you to some ill-conditioning and when you are discretizing. See we saw in finite difference method what was the message take home message that if you have more number of gird points better the solution. So when you are discretizing this is true for even for interpolation approach to discretize.

So we would need large number of grid points, but that would mean you would have to fit a large order polynomial and then you can get into ill-conditioning problem. Now when I am going to do the development of orthogonal collocation method I am not going to bother too much about this ill conditioning. I am just going to develop a method and convey the approach, but what you really need is spline interpolation.

Do you know what is spline interpolation cubic spline interpolation? Well I have discussed it in the detail in the lecture note. So I will just give you a brief idea what is cubic spline interpolation. The exact equations you can see here in section 4.2, but let me explain the idea. The basic idea is this that instead of fitting one giant polynomial of high order which passes through all the equation. We could choose to construct piecewise polynomial approximation.

(Refer Slide Time: 39:50)

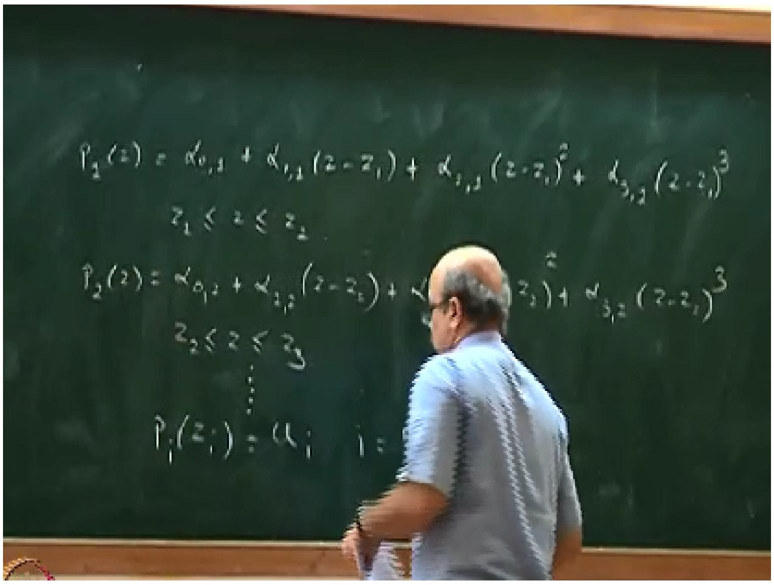


So I could construct 1 approximations between these 2 points. I could construct another approximation between these 2 points third approximation between these 2 points. What I have to make sure is that these neighboring approximations are continuous. Typically, what is done is that we fit a cubic polynomial between 2 neighboring points and make sure that there are smoothly joint.

So how do you make sure how they are smoothly joint? Well we make sure that the first derivative and the second derivatives match for the neighboring polynomials. So this is called as piecewise polynomial approximation and this is many times required because so what is the idea here is that you still have the same number of points. You still have Ui that is Uzi for ZI or I going from 1, 2 up to n+1 so you still have this grid points.

Now what I am going to do here in piecewise polynomial approximation well let me little careful about terminology piecewise polynomial interpolation approximation will be using in some other context. So piecewise polynomial interpolation so what I do here is that I fit a polynomial between 2 neighboring points and then I make sure that there is a continuity between 2 neighboring polynomial.

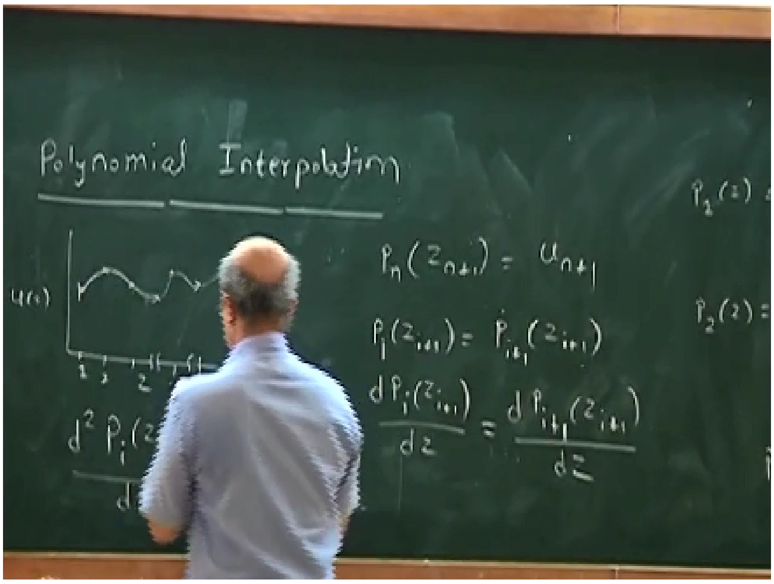
(Refer Slide Time: 41:35)



So I will have a polynomial P1z which is alpha 0 1+ alpha 1, 1. Now there are many polynomials so I have to have a notation which has 2 coefficients the coefficients will have alpha 0, 1. So this is the first polynomial 0th coefficient. Then z-z1+ alpha 21 z-z1 square+ alpha 3 1 z-z1 cube and this holds between z1 z z2 P2z is alpha 0 2+ alpha 1, 2 z-z2+ alpha 2, 2. Z-z2 square +alpha 3, 2 z-z2 cube.

So this polynomial holds between z2 z and z3. And likewise between each segment I fit a polynomial in this approach this is called piecewise polynomial approximation. Now how do I make sure there is continuity. I have to put some conditions here so one condition is of course you know there is values one condition come from this values. The other conditions come from the derivatives or the smoothness of the derivatives.

(Refer Slide Time: 43:53)



So other conditions arise so this is the terminal point so these are all initial points of each polynomial then we have these conditions which are continuity conditions P1z i+1= P i+1 z i+1. So essentially we are saying that the initial value of this polynomial is same as the final value of the previous polynomial then you have condition dpiz i+1/dz= dp i+1. And the third condition is.

So to ensure smoothness we make sure that the value of these 2 neighboring polynomials at the common point is same. The first derivative of the 2 neighboring polynomials at the common point is same and the second derivative of the 2 neighboring polynomial at the common point is identical. So these are the additional constraint which we impose if you actually put all these constraints then you need 2 boundary constraints.

So there are different ways of specifying the boundary constraints one of them is called as you just look at the notes here. So free boundary conditions are we need 2 more conditions at 2 more boundary point at this point and this point. So those are free boundary conditions would be d2p if you include these 2 boundary conditions that at the 2 boundaries the second derivative is 0.

If you include these 2 boundary conditions, then you get number of equations= number of unknowns. What are the unknown here alpha values? How many alpha values? 4*n. You can do some algebraic manipulation and finally reduce this problem to Ax=b where A is a sparse matrix. The manipulations are given here you can just have a look at it, but interpolation when it is not possible to do high order interpolation this cubic polynomial interpolation will be used.

And that will lead to orthogonal collocation on finite elements. So we will get a method finite element method. If you do polynomial interpolation on a smaller domain. So the different ways of doing interpolation and it is a very, very rich area. I am just touching a tip of the Iceberg by talking about these 2 basic ideas which I need to develop my further approach. So the next lecture, I will start using interpolation idea not the piecewise polynomial interpolation simple interpolation.

Nth order polynomial passing through all n+1 points. This simple idea I am going to use to develop method of orthogonal collocation. So by the method of orthogonal collocation we will discretize again boundary value problems and partial differential equations. So you will again revisit the same problems, but through a different approach. So orthogonal collocation is a very popular method in fact it is a method which requires less number of grid points to get the same accuracy as that of the finite difference method.

Finite difference method would require large number of grid point. This method gives better solutions with less number of grid points. So this is computationally less expensive as compared to finite difference method that is why we want to go for polynomial interpolations.