Numerical Methods and Programming

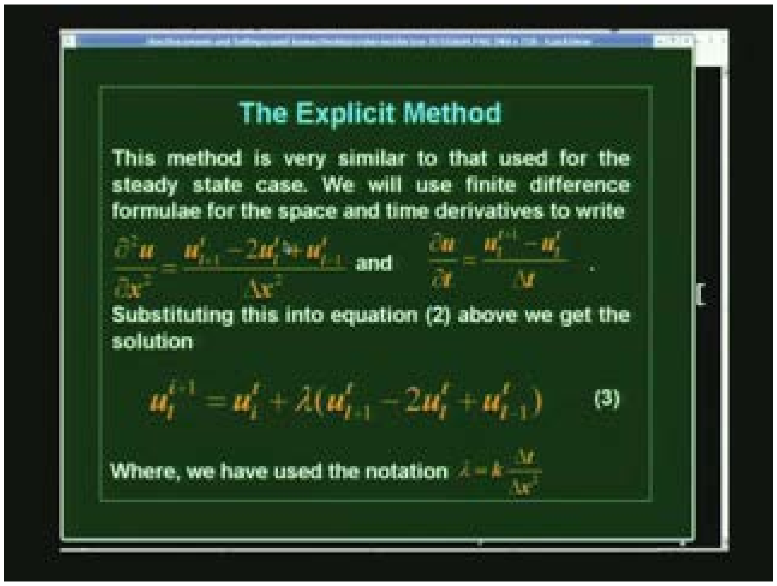
P. B. Sunil Kumar
Department of Physics
Indian Institute of Technology, Madras

Lecture - 36

The Crank – Nicholson Scheme for Two Spatial Dimensions

In the last class we were looking at solving partial differential equations using the following method. So we said that we can write the partial differential equation as a difference equation and solve it explicitly.

(Refer Slide time: 01:39)



The Explicit Method

This method is very similar to that used for the steady state case. We will use finite difference formulae for the space and time derivatives to write

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1}^t - 2u_i^t + u_{i-1}^t}{\Delta x^2} \quad \text{and} \quad \frac{\partial u}{\partial t} = \frac{u_i^{t+1} - u_i^t}{\Delta t}.$$

Substituting this into equation (2) above we get the solution

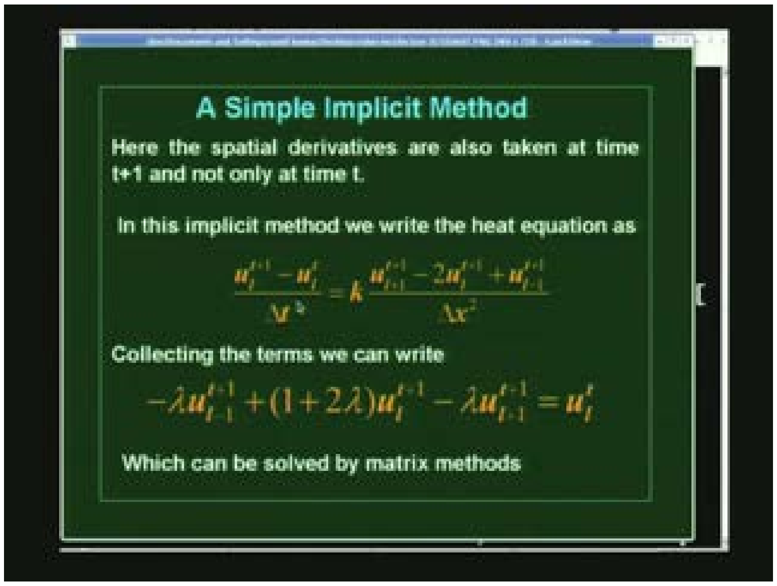
$$u_i^{t+1} = u_i^t + \lambda(u_{i+1}^t - 2u_i^t + u_{i-1}^t) \qquad (3)$$

Where, we have used the notation $\lambda = k\frac{\Delta t}{\Delta x^2}$

So using the normal numerical methods, so here in this case, the equation we are using is the heat equation that is heat conduction equation of this form, one dimensional heat conduction equation that is what you were using and then, we said that we could write the second derivative, the spatial derivative as a difference equation and use normal numerical methods to solve it, that is we can write the equation at time t plus 1 as time t plus lambda times the variables u which is actually the temperature in this particular case at time t.

So basically we are expressing the time the temperature at time t plus 1as a function of temperature at time t where lambda here, being the parameter given by k delta t divided by delta x Whole Square and we said that in this particular case the lambda has to be less than half for this equation to be stable. To get better results we went to an implicit method in which we write the right hand side of the heat equation that is, this the spatial derivative to be given by given at time t plus 1 instead of time t.

(Refer Slide time: 02:52)



A Simple Implicit Method

Here the spatial derivatives are also taken at time t+1 and not only at time t.

In this implicit method we write the heat equation as

$$\frac{u_l^{t+1}-u_l^t}{\Delta t}=k\frac{u_{l+1}^{t+1}-2u_l^{t+1}+u_{l-1}^{t+1}}{\Delta x^2}$$

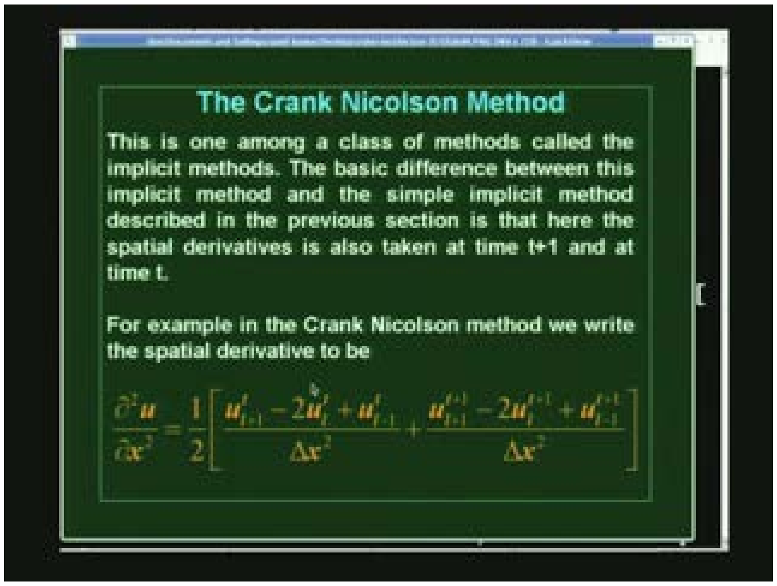
Collecting the terms we can write

$$-\lambda u_{l-1}^{t+1}+(1+2\lambda)u_l^{t+1}-\lambda u_{l+1}^{t+1}=u_l^t$$

Which can be solved by matrix methods

So then, the time derivative the first derivative is now written at the difference between the variable at time t plus 1 minus time t at 1 spatial point l here and the right hand side that is the spatial derivative now written at time t plus 1 with taking difference, using the variables at l plus 1l and l minus1 for the second derivative. Now this equations can be solved by using standard algorithms which we used for solving set of linear equations which you would write this at every lattice point and then you got a set of linear equations which can of this form and then which can be solved for u_l minus 1 at t plus 1 u_l at t plus 1, u_l plus 1 at t plus 1.

(Refer Slide time: 04:19)



The Crank Nicolson Method

This is one among a class of methods called the implicit methods. The basic difference between this implicit method and the simple implicit method described in the previous section is that here the spatial derivatives is also taken at time t+1 and at time t.

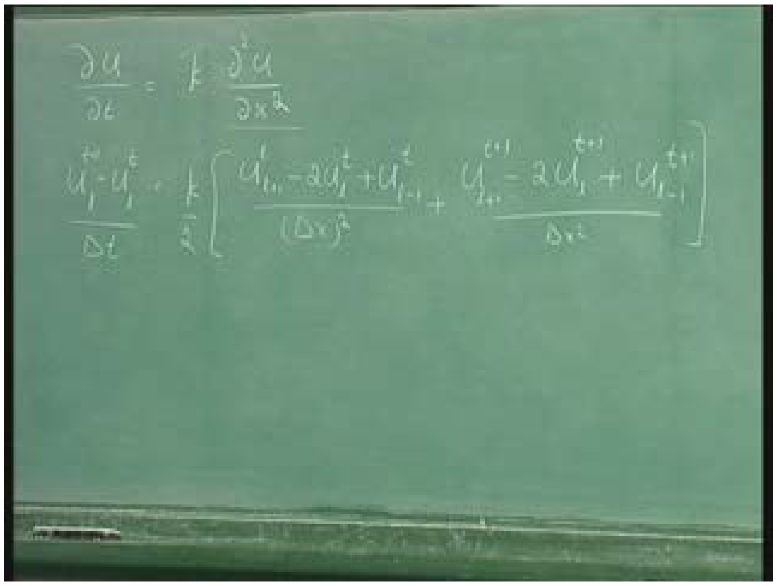
For example in the Crank Nicolson method we write the spatial derivative to be

$$\frac{\partial^2 u}{\partial x^2}=\frac{1}{2}\left[\frac{u_{l+1}^t-2u_l^t+u_{l-1}^t}{\Delta x^2}+\frac{u_{l+1}^{t+1}-2u_l^{t+1}+u_{l-1}^{t+1}}{\Delta x^2}\right]$$

So we got a tri diagonal matrix which can be inverted using standard techniques which you have learnt in the previous classes. So now, we could now we saw there is different explicit, implicit method which we call the Crank Nicholson method in which the right hand side is now written at both time t and time t plus 1that is, we take the mean of the derivatives, the second derivative of the spatial derivative at time t and time t plus 1 and we take the mean of that as the derivative. So now the equation which is del u by del t is equal to, we said lambda times, k times del square u by del x square.

So now this equation, this side is now written at both at time t plus 1 and time t. So this equation we write as our t plus 1 minus u at time t at both at the same l divided by delta t and the right hand side, you would write as u at time t_l plus 1 minus 2 u_l at time t plus u at l minus 1 at time t divided by delta x square and then plus u at time t plus 1 minus 2 u_l at time t plus 1 plus u at l minus 1 at time t plus 1 divided by delta x square with a half here. So now that is what we were writing.

(Refer Slide time: 06:00)


$\frac{\partial u}{\partial t} = k \frac{\partial^2 u}{\partial x^2}$
$\frac{u_l^{t+1} - u_l^t}{\Delta t} = \frac{k}{2}\left[\frac{u_{l+1}^t - 2u_l^t + u_{l-1}^t}{(\Delta x)^2} + \frac{u_{l+1}^{t+1} - 2u_l^{t+1} + u_{l-1}^{t+1}}{\Delta x^2} \right]$

So that equation now this equation now again has to be solved using methods which we learnt to solve linear equation, set of linear equations but now the right hand side of this equation will have a variables at that is the variable u at l plus 1l and l minus 1 at time t and that is used to determine the variable at l plus 1l and l minus 1 at time t plus 1.

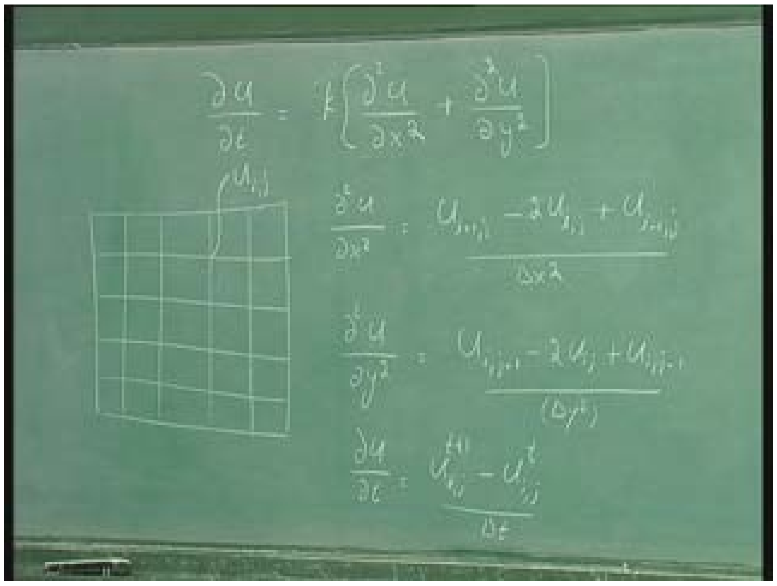
So that is the Crank Nicholson method for one dimensional heat equations. So now what we should discuss, what we will be discussing today is to extend these methods for a two dimensional heat equation that is, if you want to solve the case of heat conduction in a plane instead of just along a thin rod which we have considered here that is we will be interested in looking at equations of the form k del square u by del x square plus del square u by del y square.

So if you have to solve equations of this form then what is the method which we will use, so that will be, what we will be discussing today. So you would use similar method as the crank nicholson method that is we will split this at time t plus 1 and time t and then we would use that to solve but now we have a squared grid, so remember. So we have, so let us say we have a grid and the plate is divided into a lattice of this form. So now every point now will be represented by, we will write it in an i and j.

So at every point the temperature would be given now by $u_{i,\, j}$, now there are 2indices here because is two dimensional problems. So then we have to write these derivatives and these derivatives, so we know that we can be written as now del square u by del x square we would it as u_i represents the x here j represents the y variable. So u x y is basically written as $u_{i,\, j}$. So then now this in the discrete form we would write it as i plus 1, j minus 2 $u_{i,\, j}$ plus u_i minus 1, j divided by delta x square and del square u by del y square, we would write similarly as now $u_{i,\, j}$ plus 1 minus 2 $u_{i,\, j}$ plus u_i minus i, j minus1 divided by delta y square.

Now we could substitute these expressions into this equation and we can write del u by del t as $u_{i,\, j}$ at time t plus 1 minus $u_{i,\, j}$ at time t divided by delta t. So you can see that if you evaluate these derivatives at time t and substitute that here and then substitute this expression here and if you take delta x equal to delta y and this is exactly the same as the one dimensional explicit method and we can solve this rather easily.

(Refer Slide time: 10:03)



$$\frac{\partial u}{\partial t} = k\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right]$$

$u_{i,j}$

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}$$

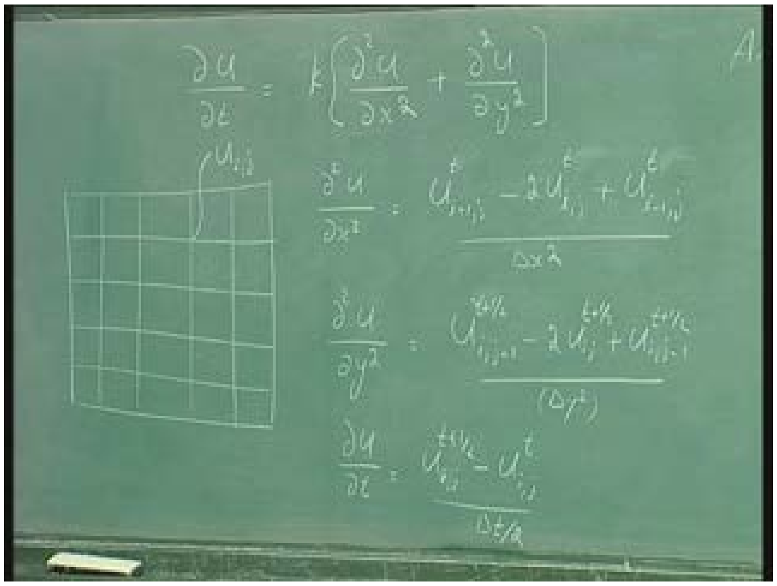
$$\frac{\partial^2 u}{\partial y^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y^2)}$$

$$\frac{\partial u}{\partial t} = \frac{u^{t+1}_{i,j} - u^{t}_{i,j}}{\Delta t}$$

So then you would get a very simple equation, you will get the right hand side to be a function of only time t and the left hand side will have both time t plus 1 and time t and you can solve this for every value every $u_{i,\, j}$ at time plus 1 in the explicit method but we know that such methods are not very stable. So we have to use very small delta x, very small lambda, so and we have seen that explicitly in the case of one dimensional equation. So what we will be interested in here will be looking at an implicit method for solving such equations and that is called the alternate direction. So that is called the alternate direction implicit, alternate direction implicit method. Okay implicit method, that is what normally known as ad_i scheme. So we will be looking at an alternate direction implicit method which basically means that we will evaluate one of the derivatives at time t plus 1 and other derivative at time t and in the next step, you will evaluate this derivative at time t and this at time t plus 1 that is the method.

So I will write that down here saying that we will evaluate this derivative, let us say at time we call it t plus half. So this we will write it as t plus half and this as t plus half and then in the second case we would evaluate these derivatives at time t and then so we will substitute this derivative at time t plus half here and this derivative at time t. Let us say, normally we do it the other way around. So I just write that we will write that as this as t plus half in the first step, we will write this as at t plus half and then we take this derivative to be t plus half minus t and delta t by 2 and then we substitute that here and then we will get an equation of the following form.

(Refer Slide time: 12:35)



$$\frac{\partial u}{\partial t} = k\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right]$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{u^t_{i+1,j} - 2u^t_{i,j} + u^t_{i-1,j}}{\Delta x^2}$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{u^{t+1/2}_{i,j+1} - 2u^{t+1/2}_{i,j} + u^{t+1/2}_{i,j-1}}{(\Delta y^2)}$$

$$\frac{\partial u}{\partial t} = \frac{u^{t+1/2}_{i,j} - u^t_{i,j}}{\Delta t/2}$$

So ad_i scheme is what we are looking at, so we will get u at time t plus half minus u at time t at i, j divided by delta t by 2. So we will write that right hand side now to be equal to two schemes that is k at now del square u by del x square evaluated at time t and del square u by del y square evaluated at time t plus half. So we substitute that here and then we will get the following equation. So if we substitute that here, so what we get is the following so we get k and we take, we will take delta x is equal to delta y. So that simplifies it and we will define lambda as delta t divided by delta x square as before. So we can substitute all that into here and then we can write $u_{i,\,j}$ at time t plus half minus $u^t_{i,j}$, we will write that as now two lambda times these equations.

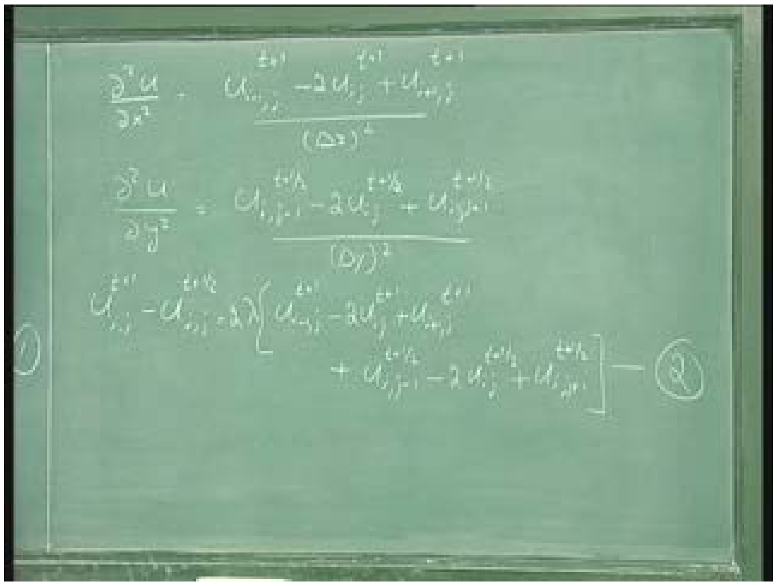
So this is evaluated remember evaluated at time t, so we will write that as u_i minus 1 at t minus 2 $u_{i,\,j}$ at, at time t plus u_i plus 1, j time t that is 1 part and then you would have the other one that is plus u now i, j minus 1at time t plus half then minus $2u_{i,\,j}$ at time t plus half and plus $u_{i,\,j}$ minus 1 at time t plus half.

So that is the equation which we have so now we will have this equation for every i and j. So which we can write and so every i and j we have this equation as $u_{i,\,j}$ at time t plus all these terms which comes on the right hand side, so this is the same as this equation. So which we can solve for $u_{i,\,j}$ at time t plus half using again our methods which we have learnt for set of linear equations and we can see this most clearly when we actually program this.

So we will get back to this equation again when we look at the program how we write this is the matrix form, for the time being what I wanted to tell you is that now once we have done this step, now we have the equations and we have solved this we have the $u_{i,\,j}$'s at time t plus half for every i and j and then we have to substitute that again into this equation for the next time step, in the next time step I will use in the next time step I will be using a for the second derivative.

So we will use u del square u by del x square as now $u_{i,\,j}$ u_i minus 1, j at time t plus 1 minus 2 u_{ij} at time t plus 1 plus u_i minus 1, u_i plus 1, j at time t plus 1 divided by delta x square and for the first derivative, for the y derivative we will stick to the half, that is u_{ij} minus 1at time t plus half minus 2 u_{ij} at time t plus half plus u_{ij} plus 1at time t plus half divided by delta y square and again we are using delta x equal to delta y that is lambda. So we get the solutions at half time that is $u_{i,\,j}$'s at time t plus half by solving this equation, right.

(Refer Slide time: 19:48)



$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i-1,j}^{t+1} - 2u_{i,j}^{t+1} + u_{i+1,j}^{t+1}}{(\Delta x)^2}$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{u_{i,j-1}^{t+1/2} - 2u_{i,j}^{t+1/2} + u_{i,j+1}^{t+1/2}}{(\Delta y)^2}$$

$$u_{i,j}^{t+1} - u_{i,j}^{t+1/2} = \lambda\left[u_{i-1,j}^{t+1} - 2u_{i,j}^{t+1} + u_{i+1,j}^{t+1} + u_{i,j-1}^{t+1/2} - 2u_{i,j}^{t+1/2} + u_{i,j+1}^{t+1/2}\right] \quad \text{— } \textcircled{2}$$

$\textcircled{1}$

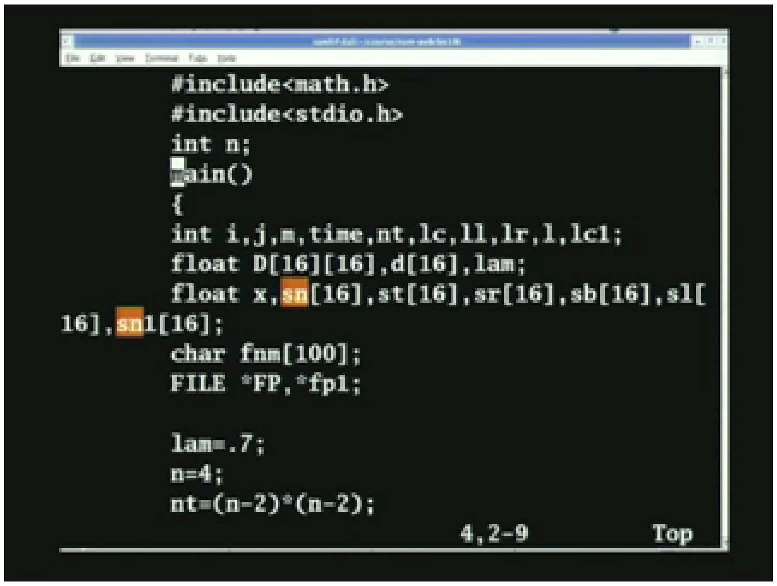
So this equation, we solve equation one let us call it we solve this equation one and substitute that and use that to solve the next step in which we will be writing u at time t plus 1 at i, j minus u at time t plus half at i, j as lambda 2 lambda times u_i minus 1, j at time t plus 1 minus 2 u_{ij} at time t plus 1 plus u_i plus 1, j at time t plus 1plus u $_{i, j}$ minus 1 at time t plus half minus 2 $u_{i, j}$ at time t plus half plus $u_{i, j}$ plus 1 at time t plus half which is equation 2.

So we solve equation one first and get the u at t plus half by using the y derivative evaluated in the implicit way while the x derivative evaluated in the implicit way and then we put that, use that and now get the and use now the equation again the same equation, again but now the x derivative evaluated at the implicit way and the y derivative in the explicit way.

So we will alternate in the direction that is in this lattice first we go in this way using the y derivative in the implicit way and then next time we will go this way using the x derivative in the implicit way. So at the end of these two time steps doing the first half and then the next half in the ones with the y derivative in the explicit way and the second case, the x derivative in the implicit way implicit way, we get the whole equation in one time step. So two half time steps for one.

So we will see how this can be how this can be written both these equations that is equation 1 and equation 2 can be written are a set of equations for each ij, we have to write this, so it is a set of equations and both these equations can be written in the form of a tri diagonal matrix equation and that is what we would be looking at and we will look at that when we are writing the program in this particular case. So let us look at a program which actually implements this case and then you have a much clearer idea of what is going on.

(Refer Slide time: 21:11)


```
        #include<math.h>
        #include<stdio.h>
        int n;
        main()
        {
        int i,j,m,time,nt,lc,ll,lr,l,lc1;
        float D[16][16],d[16],lam;
        float x,sn[16],st[16],sr[16],sb[16],sl[
16],sn1[16];
        char fnm[100];
        FILE *FP,*fp1;

        lam=.7;
        n=4;
        nt=(n-2)*(n-2);
                                   4,2-9          Top
```

So here is the program, so we need now first of all we need to write this in a, we have to put in a boundary condition, so let us put the boundary conditions which I have used here as we have the temperature set at 100 here and here it is set at 0 and the left side it is set at 75 and it is set 50 here. Okay, so all these nodes are fixed to be, so all these boundary nodes are fixed at that temperature given here and let us assume that to start with we had all the other nodes at 0, all the other nodes are kept at 0, let us assume that and then what we wan to look at is how does the temperature in the interior nodes evolve as a function of time.

So now see when we go to first in this now what we do is we set up all the other thing equal to 0 and except the boundary nodes as put at this value. So that is the first step which is done here. So this is the top nodes, okay so I put that as 100 and the bottom nodes are set at 0 and the left side nodes are set at 75 and the right side nodes are set at 50, now you see we have n nodes, okay.

So that is it goes from 0 to n minus 1 there are n nodes and we actually have to solve only the interior nodes that is n minus 2 into n minus 2, so we have the number of nodes at which we have to solve this particular equation is n minus 2 times n minus 2 and we set some lambda and I choose some n and n this case to be 4. We could choose anything this is just for demonstration.

We choose a small lattice we basically have only 2 by 2 matrix to solve the interior nodes but this program is general enough you could set this into any value if you change the dimensions appropriately. So we have that is the boundary nodes set and then now the question is how do we write this equation in the form of a tri diagonal matrix. So remember that first we have to solve for $u_{i,\,j}$ minus 1, u $_{i,\,j}$ and $u_{i,\,j}$ plus 1 here. So you have to solve for $u_{i,\,j}$ minus 1, $u_{i,\,j}$ and $u_{i,\,j}$ plus 1at time t plus half.

So now the equation when we write we have to write these equations in this matrix form, you would be writing in this. So let us take one set this is rather confusing so please pay attention. So this equation here has to be solve for $u_{i,\,j}$ minus 1, $u_{i,\,j}$ and $u_{i,\,j}$ plus 1 at time t plus half that is what we have to write. So this can be written in a simple matrix form, so then we have a matrix equation, so that equation would be of this form.

So if you have to solve just one of this equation let us say we are solving just one of these equations and then I would have $u_{i,\,j}$ minus 1 at time t plus half and $u_{i,\,j}$ at time t plus, $u_{i,\,j}$ at time t plus half and $u_{i,\,j}$ plus 1 at time t plus half that is what I would have that is equal to something on the right side that is the kind of equation which I will have one of these equations just one set of this if I write correct but then I have a series of equations of this form. So I have to solve for all the u_{ij}'s, all the u_{ij} minus 1j, j plus 1 for all i's. So the question is how do we write, so we construct a new vector which has to be solved which I call in this program as s, I call it as in this program as s_t.

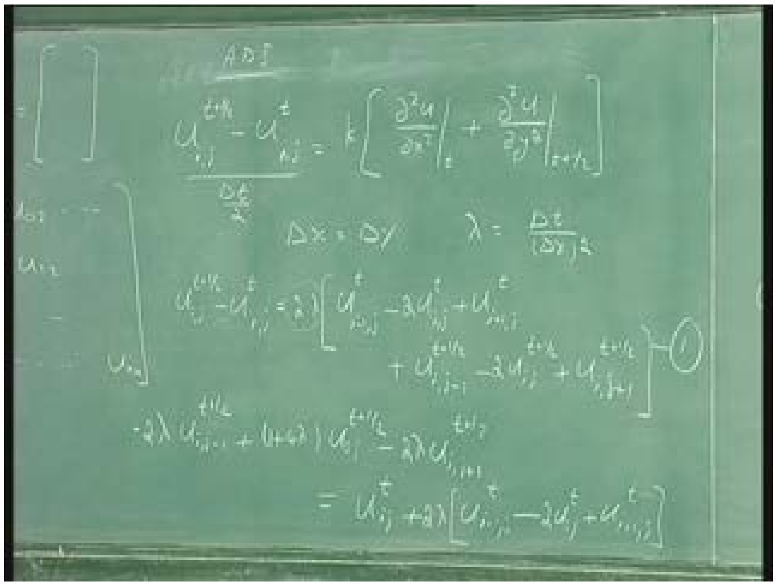
I call it as s_n in this program, so that is the new set of u values let me use the same symbol here. So this s_n is it goes like that s_n starts with, so s_n matrix is something like this. It is a column vector, so this column vector is like this. So you start with u_0 you start with u_{10} we do not have to solve the interior nodes so it starts with 0 here so we have to only solve for the interior nodes so we, the s n matrix starts from here basically and I go like this. So I will be going like this because I am doing the, I will be going down like this because I am doing only the y derivatives so when you write the matrix $u_{i,\,j}$ matrix. So let us write the $u_{i,\,j}$ matrix first.

So if I write the u matrix it will go like this so u_1, so it will go to u_{00}, u_{01}, u_{02} etcetera and then u_{10}, u_{11}, u_{12} and u $_{n0}$, u_{n1} and finally u_{nn} right. So that is what our u matrix would be so now we construct a new s matrix which actually reads it off like that is, we have a s n which I call that is a column vector which reads it off like this that is it will be u_{00}, u_{01}, u_{02} all the way down and the again then u_{10} etcetera so end with u and it ends with u_{nn}.

So u_{10}, u_{11} and then ends with u_{nn} at the end. So we have a column vector like this, so which goes like this. So it reads off u_{00}, u_{01}, u_{02} up to u_{0n} and then comes back and says u_{10}, u_{11}, u_{12} etcetera. So we have column vector which is I call s_n which is constructed in this fashion and then I will write one using this I would write just one matrix equation for this whole set and that is what I am trying to do here. So let me just show that here, so we have this boundary matrices here values and then I am constructing now I have to construct the matrix, this matrix that is what internal matrix is and that can be constructed easily from here. So let us just do that, so if you write this in a matrix form you would write this as $u_{i,\,j}$ minus 1 at t plus half that is this term, so it is minus 2 lambda. So let us call this 2 lambda as some lambda.

So we call that as lambda and then we set this as minus lambda times here minus 2 lambda in this particular case and then we have u_{ij}. So there are 2 parts, one is coming from this and one from this. So I have 1plus 4 lambda times $u_{i,\,j}$ at time t plus half and then I have similarly minus 2 lambda times $u_{i,\,j}$ plus 1 at time t plus half. So that is my left hand side and the right hand side of this equation is then $u_{i,\,j}$ at time t plus 2 lambda times u_i minus 1, j at time t minus 2 $u_{i,\,j}$ at time t plus u_i plus 1j at time t.

(Refer Slide time: 30:28)



ADI

$$\frac{u_{i,j}^{t+1/2} - u_{i,j}^{t}}{\frac{\Delta t}{2}} = k\left[\left.\frac{\partial^2 u}{\partial x^2}\right|_t + \left.\frac{\partial^2 u}{\partial y^2}\right|_{t+1/2}\right]$$

$$\Delta x = \Delta y \qquad \lambda = \frac{\Delta t}{(\Delta x)^2}$$

$$u_{i,j}^{t+1/2} - u_{i,j}^{t} = 2\lambda\left[u_{i-1,j}^{t} - 2u_{i,j}^{t} + u_{i+1,j}^{t} + u_{i,j-1}^{t+1/2} - 2u_{i,j}^{t+1/2} + u_{i,j+1}^{t+1/2}\right] \quad (1)$$

$$-2\lambda u_{i,j-1}^{t+1/2} + (1+4\lambda)u_{i,j}^{t+1/2} - 2\lambda u_{i,j+1}^{t+1/2} = u_{i,j}^{t} + 2\lambda\left[u_{i-1,j}^{t} - 2u_{i,j}^{t} + u_{i+1,j}^{t}\right]$$

That is the equation which I have to write. So the matrix elements of this equation of this matrix equation, the matrix elements when I use the whole column vector of this form on the right hand side would be all this given by this. So that will have minus 2 lambda and the diagonal elements are all 1 plus 4 lambda and then the upper diagonal element is minus 2 lambda. So you can see it is a tri diagonal matrix right it is a tri diagonal matrix provided I write this whole thing in a column vector form, this $u_{i,\,j}$ minus 1 $u_{i,\,j}$ $u_{i,\,j}$ plus 1 in the column vector form that is I have to read off I have to make a column vector which reads it like this.

So it reads off like this and then comes back here and reads off like that and then comes back here reads off like this etcetera. So if I write all the $u_{i,\,j}$ minus 1 $u_{i,\,j}$'s and $u_{i,\,j}$ plus 1 in a column vector of this form and then I have the lower diagonal elements of the matrix as minus 2 lambda the diagonal elements as 1 plus 4 lambda and the upper diagonal elements as minus 2 lambda. So I have a tri diagonal matrix and the right hand side this column vector would be given by this value this evaluated at time t that is what we will have and that is what we are constructing here. So this matrix elements $d_{i,\,j}$ remember, are not functions of u right. So they are not functions of u they are simply minus lambda.

So in this program I had set the value, what I set is the value is 2 lambda as what I call lam is actually 2 lambda because it only appears as multiples of 2 lambda. So I can call, what I call lam in the program is just this 2 lambda. So the lower diagonal matrixes are minus lam and the diagonal matrix is 2 times 1 plus lam and the upper diagonal matrix is minus lam. So now you see how I go about this to construct this matrix i go from, i going from 1 to n minus 2 I have only n minus 2 elements and j goes from 1 to n minus 2 and I just have an index called i here, l here and the l keeps increasing.

So only I have to construct the diagonal and the lower diagonal and the upper diagonal matrix, of course I have to treat the first row and the last row rather carefully because in the first row, when in the first row that is when I am actually constructing the equation for this particular point, the left hand side of this thing and the upper side of this as to be, as my special points my boundary points. So that has to be treated separately.

So that is when I am using this equation, so when I am treating this equation rather this equation so then I have the u_j minus 1 term coming in here that is when I am using this. So u_j minus 1, so that is actually the point not j minus 1, j plus 1 that will be the point which is actually here which is actually fixed. So that does not, that goes on to the right hand side of this equation because that is a known quantity it is all temperature it is fixed that goes on to the right hand side of the equation.

So we have only these 2 points, so that is the thing which has to be taken care of carefully that is what is done here so these special points l equal to 0 and l equal to n minus 2 are taken as special points. Okay then we construct this matrix once we have constructed that matrix and we find the inverse of this matrix and we have gone through this earlier how do we find the inverse of this matrix.

So I have a program we have, I have a function here which computes the inverse of this matrix and then here is the function here is the function here is the inverse of the matrix and returns it back to the main program. So once we call this program the d will be the matrix d will be replaced by its inverse, so you remember this is something you have seen in the earlier part of this lecture series that when you can pass a two dimensional array as an array of pointers. So this has been passed here as an array of pointers and this program actually replace the d with its inverse so what, when it calls so what it returns here is actually the inverse of this matrix.

So now d is now the inverse of that matrix which we had just called. So now once we have the inverse of this matrix now this part of this program is actually constructing that column vector which we just now said. So first I initialize that column vector, so it goes from 0 to n minus 2 times n minus 2 actually minus 1 there should be a minus 1 here. So this is another thing which is rather confusing when you do the c programming. So you start, all the arrays start with 0.

So we had n elements and then of which the two boundary points were fixed. So we have a n by n vector n by n matrix here of which the boundary points are fixed, so we actually have only a n minus 2 by n minus 1 by n minus 1, sub lattice which we have to the n minus one by n minus 1 sub lattice on which we have to solve this, of which then when you dimension it when you write the array it actually goes from 0 to n minus 2. So number of elements will be n minus 2 times n minus 2 minus 1 and then, and then we construct this matrix in 2 sets right, so first we have to make this matrix here.

So to make this matrix I use this one, so let us forget this particular line for the time being it comes in the second time step. So now first time step let us say time step one what we need to do is to construct this matrix s_n the column vector. Okay so to start with s_n and s_{n0} are, s_{n1} are both 0s if everything is 0.

So then the way we construct that is actually say I minus 1 times n minus 2 plus j minus 1 now these minus 1 remember j is the y index and i is the x index now these minus 1 appear here only because it is a c program and everything starts with 0. So and there are n minus elements in each array. So this n minus 1 appears here because it is a c thing and it starts with 0. So the first, when I say I equal to 1, j equal to 1 it should actually the 0 th element, not the first element that will be, so that is what it will give.

So, i equal to 1, j equal to 1 means it is 0 l, c_0 and then j equal to 2 would give me s equal to 1 and when j goes to n minus 2 we have n minus third element. So that is what it is, so we are taking this quantity here we start with 0, 1, 2 so when n is, so we have and then once this reaches the n minus 1 and then we switch the first element and goes to the next one that is first you vary you run along all j's and then you start with the next element of u. So let me write it more explicitly here, so let us say we have only 0, 1, 2 and then the next 1 would be u_{01}, u_{11} and u_{12}. So it will keep going like that and that is what we are doing here.

So it is lc, so it is actually constructing this matrix to start with all this is initialized to 0, all interior points are initialized to 0, as I said earlier. So that is done outside the time loop, outside the lime loop we initialize everything to 0, so let us only worry about s_n right now and lc and then we have this the right hand of this is to be constructed of this matrix now the right hand side of this matrix equation has to be constructed in similar fashion. So using this one again.

So remember again we have to take the i minus 1 and i and i plus 1 and you have to be careful, when you do that because again the boundary points has to be treated correctly. So when i is 0 and i is n we have to treat these 2 points separately more carefully and that is what is done here. So the right hand side of the matrix is lambda times s_{ll} and 2 into 1 minus lambda times that is what we have 2 lambda times and then 2 into 1 minus lambda.

So that is we can combine these two and the 2 lambda is lambda here remember, so it is 2 into 1 minus lambda this combining $u_{i,\ j}$ t and here $u^t{}_{i,\ j}$ here. So that is this part this second part and then the last part is lambda times this one is right so this is the, so I call i the right as now again, we remember from the i and the j we have to construct this l. So that is what I am doing here.

So I am doing using the same thing which we have used here that is i minus 1 times, n minus 2 plus j minus 1. So that determines which is the l value and then, so I am just doing the i minus 1, n minus2 j minus 1that is the actually lc equal to l center and then I have to take the left and the right of that that is i minus 2 and i. So that is the values I am using here. So i minus 1 value i minus i, i minus 1 and i minus 2 that is what we are using and that we use in the this program here. So this is nothing but just writing this equation here. So I am just writing this particular equations remember that.

So I have i minus 1j, i plus 1 j and i j and I have to determine the corresponding number corresponding element of this column. So which column like this column we also have a d right hand side column here which I call d there and that also goes as d actually for this one if I call by ij should go as d_{00}, d_{01}, d_{02} and d_{10}, d_{11}, d_{12} etcetera.

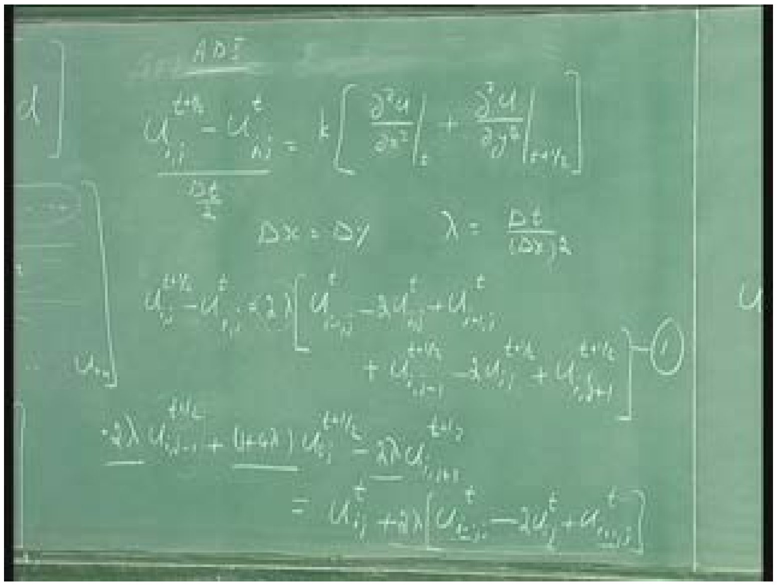
So that again has to be written in this series hence I have to determine the corresponding element here that is the conversion into the column vector from the two dimensional matrix I want to make it into a column vector and that is what is done here. So this is the center part and this is ij and this is i plus 1, j and this is i minus 1j. So i minus j becomes i minus 2 times n minus 2 ij will become i is i minus 1 times n minus 2 into j minus 1and i j is i plus j is i times n minus 2 plus j that is what is substituted here and again the points i equal to 1.

So that is for the general interior points and for the point inside for the boundary i equal to 1 and i equal to n minus 2 has to be treated specially because there are boundary points this is the right point come here and the and the left point comes here etcetera remember this is our s_l and s_r here are the boundary points that comes at i equal to 1and i equal to n minus 2 it has to be treated specially.

So once we have done that so once we have constructed the right hand side and the left hand side of this matrix equation. So we can solve this now remember d is now already the inverse of that matrix. So we have constructed the whole left hand side and the right hand side of this thing and we also found the inverse of this matrix. So we know now we can determine this at time t plus 2 by multiplying the right hand side vector here which is d by the inverse of this matrix m.

So this is d and this is now m and this is s our matrix in particular this is capital D and this is our s_n matrix this one. So we take the d inverse and multiply it to the d inverse and then we write s_n which is the value of u at time t plus half remember it is at half because we have used only the y derivatives the y derivative at time t plus half.

(Refer Slide time: 44:39)



ADI

$$\frac{u_{i,j}^{t+1/2} - u_{i,j}^{t}}{\frac{\Delta t}{2}} = k\left[\frac{\partial^2 u}{\partial x^2}\Big|_t + \frac{\partial^2 u}{\partial y^2}\Big|_{t+1/2}\right]$$

$$\Delta x = \Delta y \qquad \lambda = \frac{\Delta t}{(\Delta x)^2}$$

$$u_{i,j}^{t+1/2} - u_{i,j}^{t} = 2\lambda\left[u_{i-1,j}^{t} - 2u_{i,j}^{t} + u_{i+1,j}^{t} + u_{i,j-1}^{t+1/2} - 2u_{i,j}^{t+1/2} + u_{i,j+1}^{t+1/2}\right] \quad \text{①}$$

$$\underline{-2\lambda}\, u_{i,j-1}^{t+1/2} + \underline{(1+4\lambda)}\, u_{i,j}^{t+1/2} \underline{-2\lambda}\, u_{i,j+1}^{t+1/2} = u_{i,j}^{t} + 2\lambda\left[u_{i-1,j}^{t} - 2u_{i,j}^{t} + u_{i+1,j}^{t}\right]$$

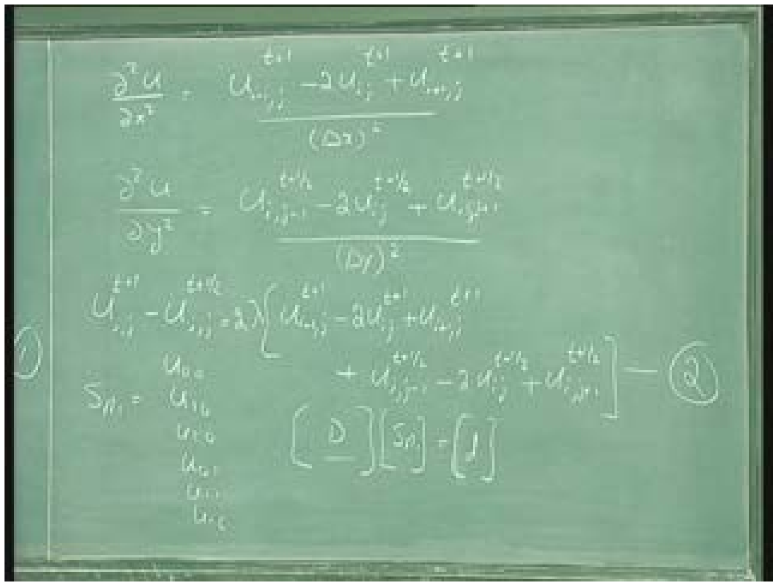
So we have only constructed the y derivative in the implicit way, so we have only we are still taking the x derivative at the explicit way in this method that is the first part of this equation. So we have got the new s_n values by multiplying this matrix d this little d by the upper case d here, which is the two dimensional matrix which is the inverse of the original matrix now.

So once you have the new s_n values we have constructed all the values at half time. So now we have all the values at half time, so now we go to this particular equation now this equation can also be written just like what we had written in case of equation one we can write this equation also as a tri diagonal matrix provided, we now we are able to write u_i minus 1j, $u_{i,\,j}$ at u_i plus 1j as a column vector now our s_n matrix now which I call s_{n1} in this case in the program is now should go like this that is it should go as u_{00} now it should go as u_{10}, u_{20} and then I should change u_{01} and u_{11}, u_{12} etcetera.

So now that is the difference, so s_n was u_{00} ,u_{01} ,u_{02} and u_{10}, u_{11}, u_{12} etcetera and now s_{n1} is u_{00}, u_{10}, u_{20} etcetera. So the I index changes first keeping the j fixed because we want to write this as a column vector. So now we want to write this part also as a matrix which is d and times now s_{n1} and that is equal to again right hand side which I call d now, you can see that this is the same as earlier one which if I write this I will get minus lambda and 1 plus 2 lambda and minus lambda.

So that is what you will get. So you will get $u_{i, j}$ u_i minus 1 j at time t plus 1 we got a 2 lambda there and a $u_{i, j}$ at time t plus 1. We will have one contribution from here and 1contribution from here there is 1 plus 2 lambda and then you have a u_i plus 1 at j which is having a 2 lambda. So that is exactly the same as equation one if you look at this it will be exactly the same as this, no difference only thing instead of j's now it is i's which I vary here.

(Refer Slide time: 47:36)



$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i-1,j}^{t+1} - 2u_{i,j}^{t+1} + u_{i+1,j}^{t+1}}{(\Delta x)^2}$
$\frac{\partial^2 u}{\partial y^2} = \frac{u_{i,j-1}^{t+1/2} - 2u_{i,j}^{t+1/2} + u_{i,j+1}^{t+1/2}}{(\Delta y)^2}$
$u_{i,j}^{t+1} - u_{i,j}^{t+1/2} = 2\lambda\left[u_{i-1,j}^{t+1} - 2u_{i,j}^{t+1} + u_{i+1,j}^{t+1} + u_{i,j-1}^{t+1/2} - 2u_{i,j}^{t+1/2} + u_{i,j+1}^{t+1/2}\right]$ — 2
1
$S_{n_1} =$ u_{00} u_{10} u_{20} u_{01} u_{11} u_{21}
$[D][S_{n_1}] = [d]$

So we can write this matrix in this form only thing we have to change is we have to change the s_n values now that has to organized in the column vector of this form that is now u_{00}, u_{10}, u_{20}, u_{01}, u_{11}, u_{12} etcetera that is important.

So but now remember we have got the values which we are going to use on this side
that is for this quantity also has to run the same way. Okay so we have to, remember this thing which we have got now the half time step u values, we got as s_n we have got this as remember this we had got as s_n now in the s_n of l where the l was remember is actually n minus 2 times i plus j that is what it actually is because for each I, we ran over j in the column vector form.

So now that has to be changed back into this form so there is if you want to write it in this form so that is the only change which we have to do that is what I am doing here. So we change this back into the new column vector. So we have to make this change in the column vector and write the new column vector in this form. Okay so now it becomes j minus 1 times n minus 2 by n minus 1 remember earlier we were writing i minus 1 times n minus 2 into j minus 1 that is lc 1 that is what is used for sn where s_{n1} the new one will use j minus 1 times n minus 2 into i minus 1 because we are doing it in the other way around.

So now once you have that again we construct the right hand side the matrix. So now the boundary points now these points again remember these boundary points, this j equal to 1and j equal to n minus 2 again has to be treated now specially. So that will get the bottom and the top points because this j which is changed, j equal to 1 when we take the j minus 1, it begets the j equal to 1 we get the j minus 1 then it will pick up the, pick up the bottom point and when j equal to n minus 2, we get the j plus 1 that will pick up the top point.

Okay so that is the only difference between the x and the y values. So now we solve for that and then we just, so that is the matrix equations you remember this d is still the same no difference it is still the same and we use that to get the new s_n values. So once we got that, then we go back here from this again we go back here and solve for the next half time step again we remember, we have to again switch back to this column right back. So that is then we keep doing this to get the correct equation.

So that is the way to solve the alternate different, alternate direction implicit method.
So we first half, so remember to conclude what we do is the first half we descriptize let us say y at time, we use the implicit descriptized y at time t plus half that is using the implicit method while we keep the x derivatives as in the explicit form and then the next time step we would write the x derivative in the implicit form and the y derivative in the explicit form, the only complication, in principle it is very simple the only complication comes in the programming because we have to write this in the column vector form remember that.

So in the first time step the column vector runs like this as u_{00}, u_{01}, u_{02} because the second index represents the y value. So we represent we run it like this and the first time, in the next half time step we would run it downwards like that, so it will go like this that is we go from u_{00}, u_{10} to u_{n0} and go from u_{01}, u_{11}, u_{21} etcetera that is what the two s values are that is what you have to remember.

So you have this matrix equation and then you have this s_{n1} and s_n. So s_n is s_n is, so this is one thing and the other one is u_{00}, u_{01}, u_{02} and u_{10}, u_{11} etcetera. So remember this this is if you remember that if you have to write this in these two different forms when you do with two different steps that is the implicit y step and the implicit x step. We have to do this, we have to make this change over about then the programming is extremely simple the d matrix that is the left hand side matrix is the same for both of them and which can be constructed in the beginning of the program and we can find the inverse of that and keep it and you do not have to change this matrix in the time as the time proceeds that makes it extremely easy to evaluate this is the most time consuming part for large n values because this here. We have to the inverse of a large n by n matrix and that is the most time consuming part but that has to done only once like this in this case that brings to an end the part on the partial differential equations we will be looking at only these equations.

So we have looked at the elliptic equations while solving the laplace's equation and how you would solve that and then this heat equation in both one dimension and in two dimensions using implicit, explicit methods and a simple implicit method Crank Nicholson method and alternate direction implicit method which is also a modification of the crank nicholson scheme for the one dimensional case.

So that is the end of this part and in the next class, we would look at the another numerical most very often used numerical technique that is to find the Fourier transform of a series Fourier transform of a set of data points that is what we would be looking at and fourier transforms are used at various for different cases in engineering and science in fact it can be also be used for solving differential equations some linear differential equations can be written as can be converted in to a set of algebraic equations by taking the Fourier transform of that in for special cases, not for all cases but for special cases.

So Fourier transforms are another important um method which we should learn and that will be what we will be looking in the next class.