NPTEL
NATIONAL PROGRAMME ON
TECHNOLOGY ENHANCED LEARNING

IIT BOMBAY

CDEEP
IIT BOMBAY

Phase field modeling; the materials science, mathematics and computational aspects

Prof. M P Gururajan
Department of Metallurgical Engineering and materials Science, IIT Bombay

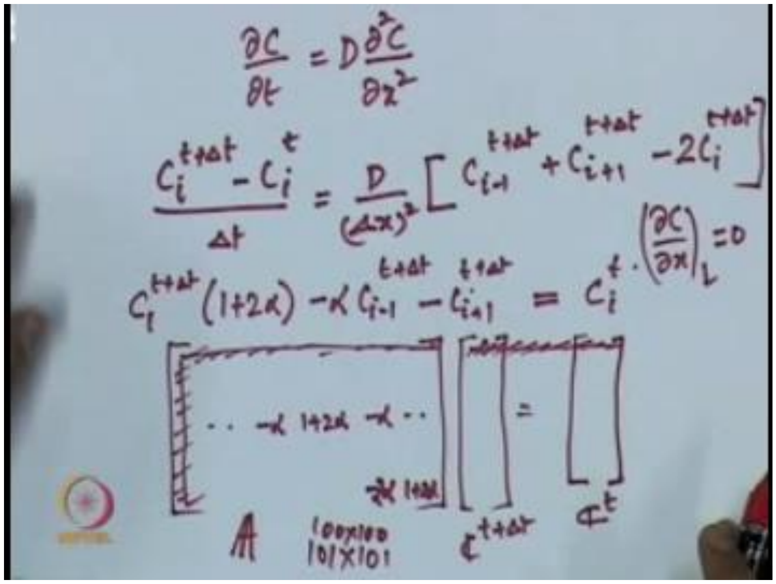
Module No.9
Lecture No.37
Diffusion equation:

zero flux BC &

implicit method

Welcome we are solving the diffusion equation numerically so we are looking at the discretization scheme.

(Refer Slide Time: 00:22)



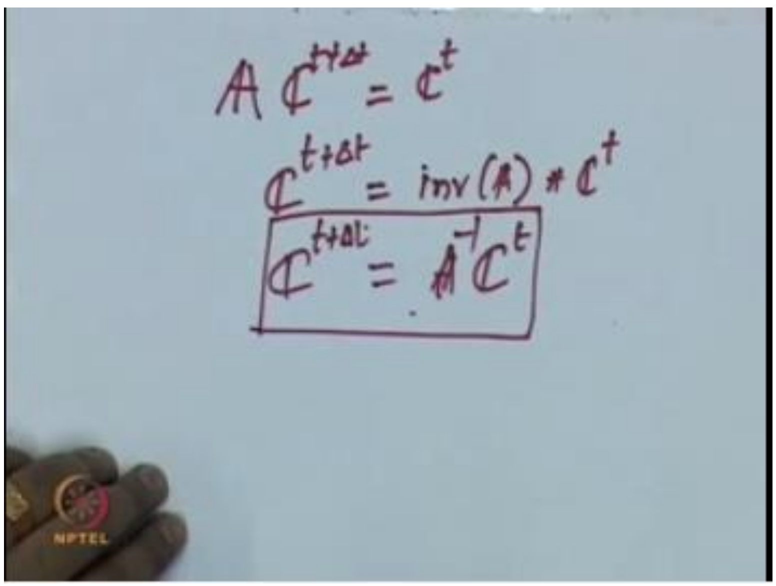
Let us look at the discretization equation that we are trying to solve okay we are trying to the implicit discretization how does that work C_i time t + Δt – C_i at time t / Δt is going to be equal to D/ Δx^2 and the we have $c^t{}_{i+1}$ + Δt + $C^t{}_{i+1}$ + Δt -2C_it+ Δt so everything is at t+ Δt so that is by this is a implicit scheme so multiply Δt here by Δx^2 that quantity is what we are going to call is α so we have C_i^t+ Δt multiplied by 1+2α and then – α times $c^t{}_{i+1}$ + Δt – $c^t{}_{i+1}$ + Δt is going to be equal to C_i^t of course it is written vectorially.

So you will have this Cat time t vector is equal to C at time t + Δt vector that will be multiplied by an matrix and the matrix terms look like – α 1+2 α and – α so general so we are not going have the first point line at all so this we are going to remove because we know that 1t + Δt and C_1^t is t is a same this value remains unchanged so we are going to so we if you have 101 / 101 that is what we have initially because this is 101/ 1 this is 101/ 1.

So that is going to reduce to 100/ 100 right because 1 row 1 column is going to way from everywhere if this also becomes 100/ 1 so that is why 100/ 100 is what is that we still have a problem because when you come here so 1+2 α you have a – α to the right of it which is not there but you have – α here that I going to become – α why because remember when we did this implementation of $\partial c / \partial x$ at L =0 that implied that C_{i-1} and C_{i+1} at this point is the same so say i+1 was replaced by C_{i-1} so it becomes so two time α time C_{i-1} just for that last point.

So only for the last point that is if you have 100/ 100 matrix then this is the 100^{th} row 99 column will be -2 α instead of just – α because there is no point beyond 1+2 α in the other the diagonal term is 1+2 α to the right and to the left we will have – α so if you do this let me call this ,matrix as some A matrix then you have in terms of matrix equation the following equation what do we have we have the matrix equation which goes something like this it says.

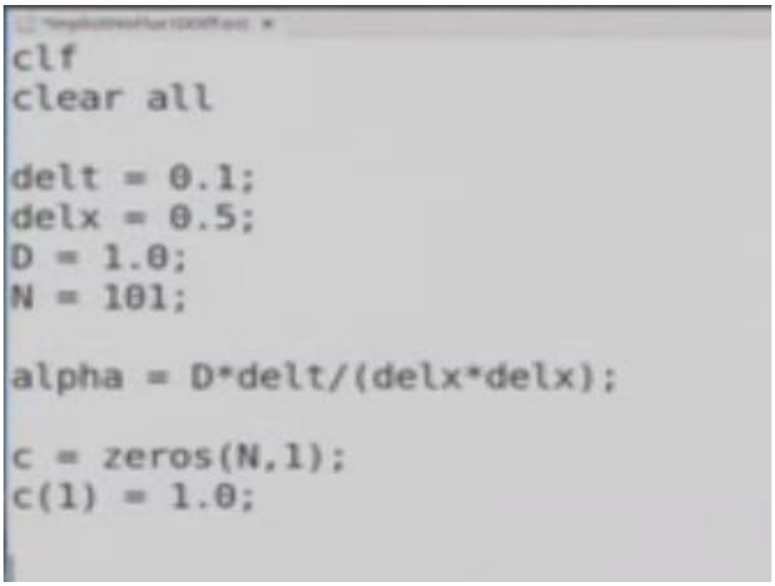
(Refer Slide Time: 04:06)



A matrix multiplying $c^t + \Delta t = c^t$ so if I want to $c^t + \Delta t$ all I need to do is inverse of A and that should left multiply this C t this is the matrix multiplication right so this is typically written as A^{-1} c^t so that will give $c^t + \Delta t$ okay this lines that I am drawing is to indicate that these what it is that we are talking about all matrices they are not just numbers okay so this is what we are going to implement numerically and you see that inversion of A is very simple and there is also some short cut notation to do these thing.

Because otherwise if you are writing your own code this is the key part of this all where inverting a matrix and left multiplying it to the right hand side to get the solution but that is going to be fairly trivial when we are looking after so let us write a script in octave to solve implicit no flux only defection equation okay so I am going to as usual.

(Refer Slide Time: 05:29)

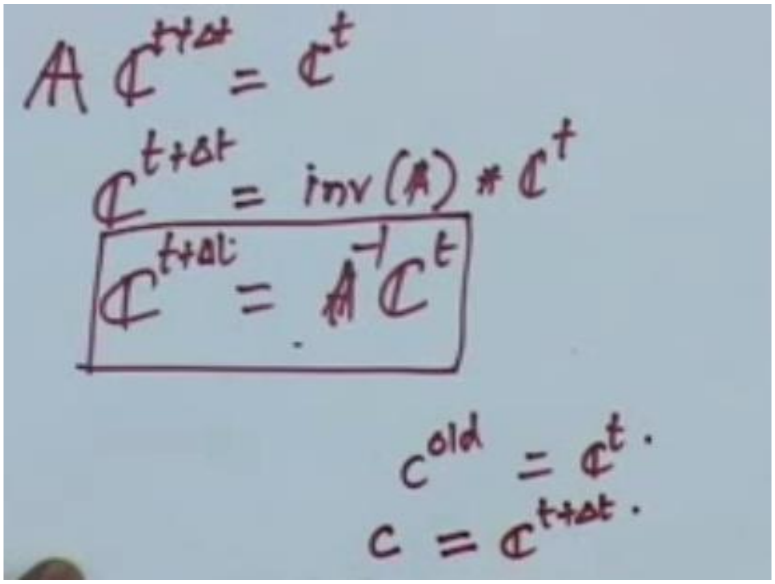


Open G editor file I am going to name the file as implicit no flux only diffusion okay file new so I am going to save as I am going to change just that right explicit I am going to change as implicit right implicit no flux 1D diffusion .oct that is what the new file is okay so we are going to do the something as we did earlier so the first thing to do is to clear all the figure and clear all the variables that might be there because I used to hold on now if I go run the script that the figure from the other script is still there this is also going to plot of the same plot which I do not want okay.

So I want to start it a fresh so I am going to start let us say clf is the first clear all that is second step and then as we did earlier I am going to defined some del t = 0.1 del x = 0.5 the reason why I am rewriting this and I am talking the older file and edit it is because if you write it a few time then it is better you get better practice so I am going to write but later we will just take the file make the modification and all that okay so for now let us take it and then are the diffusivity is taken to the unity and also going to define this quantity called N = 101, so as you know 101 is the size okay so I am going to define α, what is α? D times del t/del x * del x okay so this is α, now we are going to get the initial profile so how do we get the initial profile, we define C = zeros with N rows and 1 column.

So that is how we make the vector, remember last time we did not define this N we directly wrote 101, 1 that is possible but you can also define variables, okay. So those of you have done programming it is always a good idea to define the variables because later if you want you can just go change the N value here everything will correspondingly change, okay. So I am going to make c(1) = 1.0. So this makes the initial profile the first point has value of 1 the rest of it is 0 of course we are going to define a new variable, why is this important so let us move back and look at this a little bit more carefully.

(Refer Slide Time: 08:14)

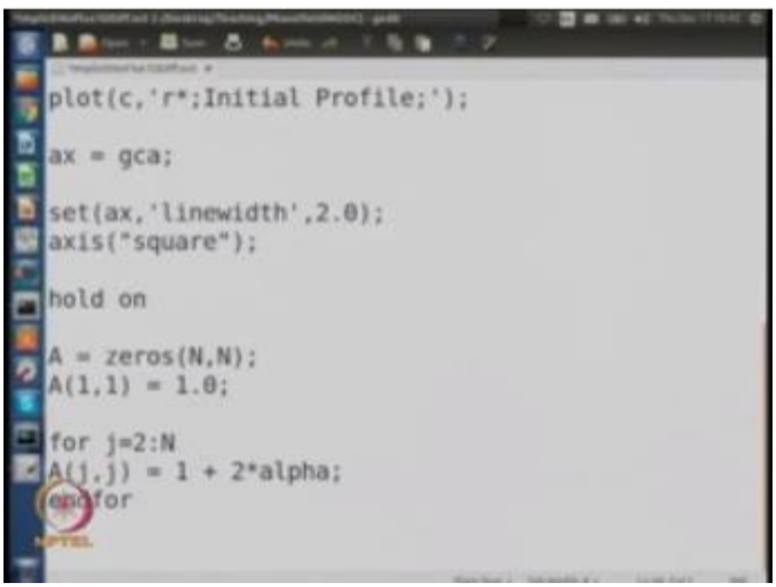


If you look at the vector equation that matrix equation that we are going to solve, see in the earlier case when we did explicit we just said C(i) = humping because the left hand side was all previous time and the right hand side was all previous time left hand side was all a new time and there was no interaction between C(i) and C(i+1) etc right, if I calculated for example I = 2 the new I = 2 was on the left hand side everything related to the whole time was on the right hand side.

So you just got the solution but when you are solving matrices then there is going to be interaction between which is old and which is new so you need to keep the distinction between which is on the whole time which is from the new time, to do that I am going to define a C old and a c we RC new, okay. So the $C^{old} = C^t$ okay the C that you are going to get is basically a C at time e+Δt.

So at every time step we are going to take the current C and we are going to make it C^{old} so that the future time step you can get the new C, this is important because in implicit scheme we are trying to do it using a matrix multiplication and then you need to keep the distinction between which is from the old time which is from the new time, in the case of the explicit it is just an algebra case expression. It does not interact with the other C's that you are calculating so you can do it using the same variable you can show the new result but here it is not possible, so let me do that so I am going to say for.

(Refer Slide Time: 09:59)



J = 1 to N, C old (j) is nothing but a C(j) end for okay, so now I am going to plot C with a red line and then I am going to call that as initial profile right, and then I want to plot it with let us this time let us not plot it with a line let us plot it with some start okay, so this is the plot okay you can notify the figure if you want so I get the handle on their axis and then I say gca set(gca, 'line width' , 2.0) right.

Set ax know ax is the gca so set ax line width is 2 and I also make axis as a square because typically I like the figures 2 have square axis, then you have to say hold on, okay. So till now whatever we have done in the earlier cases what we have done in this case also except for too small changes one is that we have defined a new variable called C old unlike in explicit case we are not using the same C to store T and T +Δt.

Because there it was done sequentially here everything is done at the same time, so you need to keep the distinguish between what is old time and what is new, second thing is while plotting the initial profile there it was plotted as a line now I am plotting it with some stars so that you will see even on the x-axis when the initial profile is plotted so that is the reason why I have done it, okay.

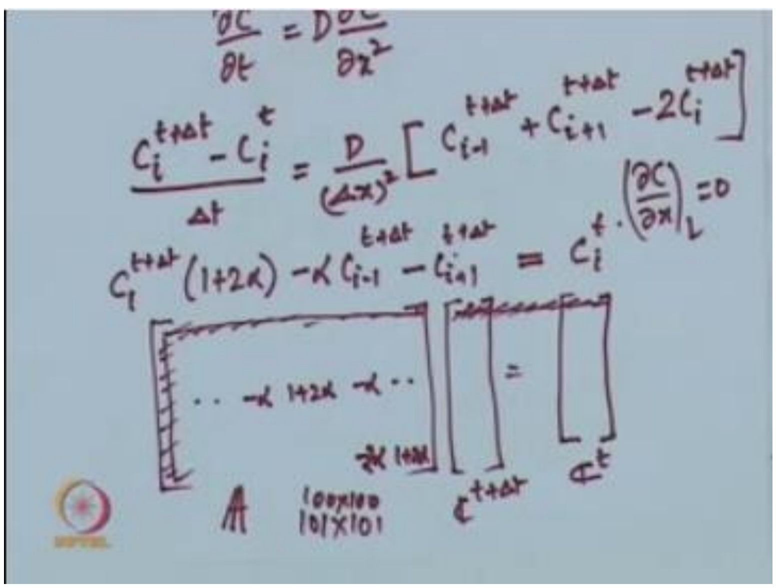
Now I am going to make the matrix so let us define the left hand side matrix so how do we do that, we say A of course works because we know that o is the way to define matrices and how big is this, this is N/N right okay and then so I said that the first point you do not need to do anything right the so I have now got A matrix which is a left hand side matrix which in what has to be inverted and multiplied the right hand side.

And it is far more easier when you are doing the matrix inverse and things like that to keep the complete matrix you know otherwise you have to get rid of the matrix and the vectors have to be changed in size and then they have to be put together for plotting the solution and all that, so I am going to avoid all this by making the first row first column the diagonal term that is the A11 to be 1.

The rest of that line is 0 in which case the action of that particular row and column on the vector is that it will leave the first term and change, okay that is what we are doing A(1,1) = 1 okay I have defined this what is this, this is nothing but the implementation of the first boundary condition namely that at location x =0 or I = 1 the composition always remains a same at 1 so we have defined the composition to be 1.

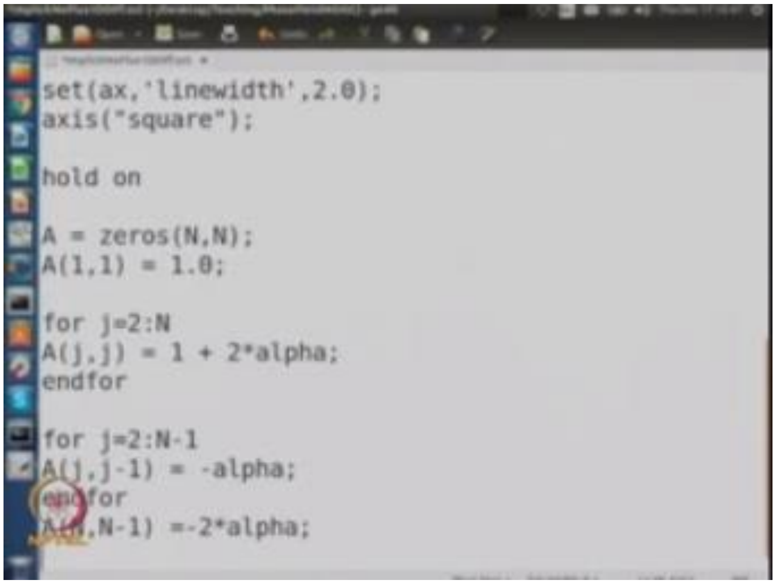
And A(1, 1) making it 1 and leaving the other terms in that first row and first column to be 0 means nothing else will change so the composition at position x = 0 will remain the same and so that is what we have done, now for the rest for j=2:N right, for j=2:N what should we do A(j,j) is nothing but 1+2*alpha, right end for, right so we are trying to define this matrix take a look at this matrix what was this matrix.

(Refer Slide Time: 14:36)



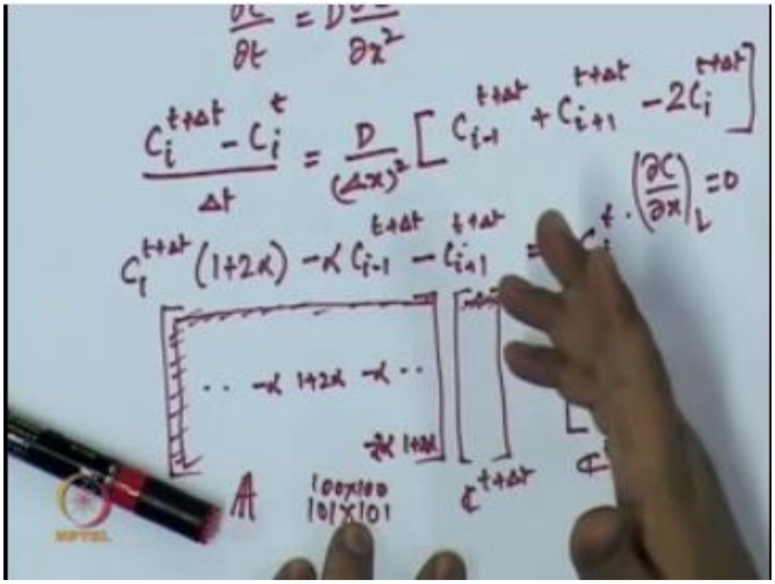
The diagonal term right, this term I made it as 1 the rest is 0, so that 1 multiplying this will be this nothing else will happen. If there are any other terms then it will mix this term with diagram because they are all 0 then 1 will just multiply this and put it here, so it is not going to do anything, right. So now all the other terms 22, 33, 44 extra should be 1+2*alpha that is what is being done in the form so I have taken j 2:N.

(Refer Slide Time: 15:04)



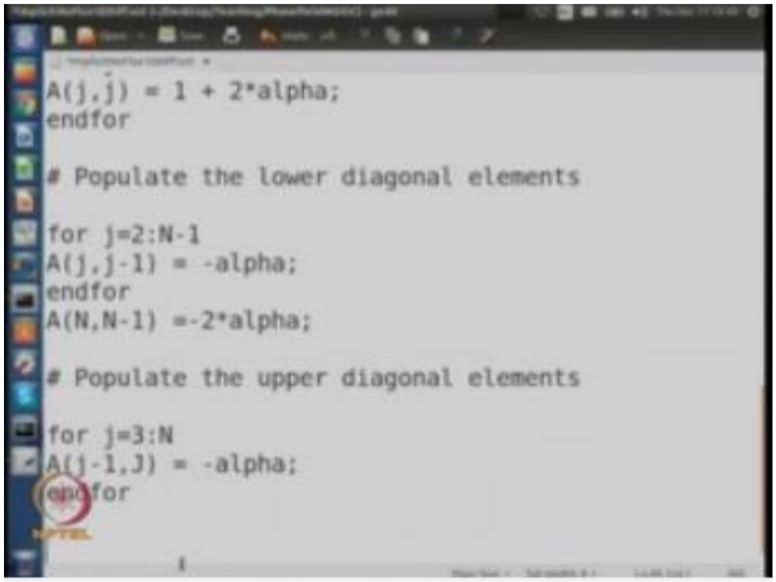
The diagonal terms A(j,j) is basically the diagonal term it is 1+2*alpha, right. So now what happens for j2:N-1, because remember when we are doing the terms to the right of the N^{th} term and left of N^{th} term there is a problem so that we want to do separately because that is the imposition of the second boundary condition for all the other points starting from 2:N-1 you can make A(j,j-1) to be equal to –beta, okay –alpha, right. So this is the terms to the left, okay except for one term which is A(N,N-1) that is equal to -2*alpha, right. So what is it that we have defined.

(Refer Slide Time: 16:23)



Again let us go back to the matrix I have, I have defined all these terms, right second one will have first and third one will have second, fourth one will have third extra, I will have –(i-1) which is –α except for the last one end for the N, N-1 is basically -2α, so that is what we have define. Now we have to go and define the upper diagonal terms, okay.

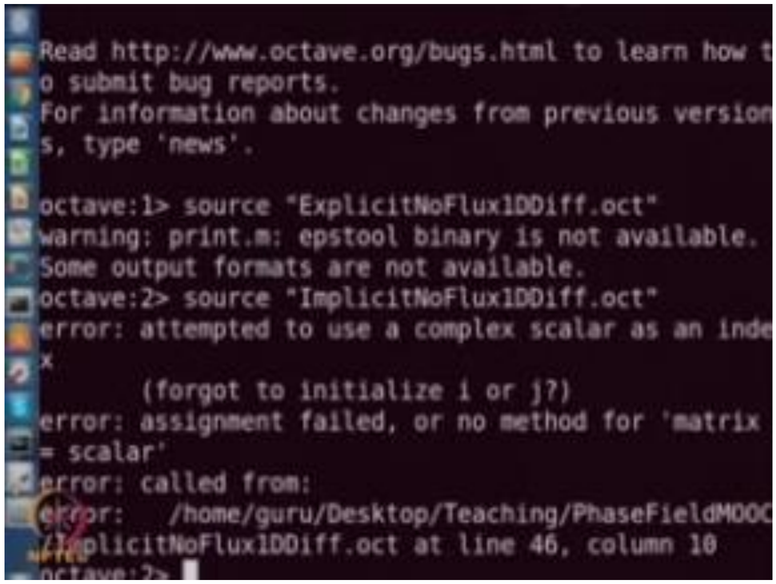
(Refer Slide Time: 16:49)



The terms to the right of the diagonal, okay so that we are going to do define the upper diagonal terms, so maybe it is a good idea to put here the commands so populate the diagonal elements, right. Now here populate the lower diagonal elements, right of the A matrix, okay. now let me populate now upper diagonal elements, okay so I am going to say for j=3:N, okay A(j-1,j)=-alpha end for, okay so all the terms to be upper diagonal so we have obtained it, and remember so it is important to know this runs from 2:N-1 and this runs from 3:N because it is n-j-1 and j so okay, so that is what the points are going to be.

So we have end for, okay now we are going use the same idea of three loops one for plotting other one for time stepping and the last one in the explicit scheme we had a loop for doing the computation itself that is not needed, so we are going to have only two loops one is for plotting, other one is for time stepping the third loop which was actually for evolving is now replaced by a matrix operation, inverse of A multiplying the C old which will give you the C in new and at the end of that we make the C new as we C old and we proceed with the iteration, so that is what we are going to do.

(Refer Slide Time: 19:02)



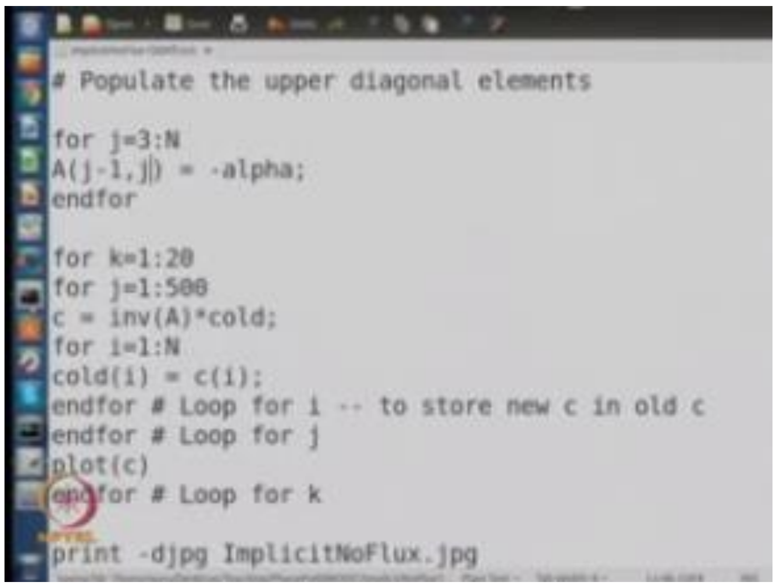
So let us do that for k=1:20 so let us plot 20, for j=1:500 so every 500 steps once we do what do we do c says c=inv(A)*cold, okay so this is a way there is also another way which is A/cold that will also get you the same solution we will talk about it little bit later. So then what do we do, so remember so loop we have not ended in this loop we first take so this is for time stepping so one time step we have done after that what do we do we say for i=1:N, cold (i) is basically c(i) end for, so whatever c I have calculated I make it as cold, so that I can go for the next step.

So this is loop for I, right which is basically to store the new value, to store new c in old c that is what this loop does. Then end for this is loop for j, right because it calculates once for j=1 for example it will do and then it will take back to be the old c so it will go, it will do 2 another time step is done it will save that new c as the old so that it can go to the third time step and so on and so far.

And then of course we have to plot we have to plots C and we have to end the last low so this is the loop for k that the end and of course we print a plot print –d so let me plot this save this 5 less d jpg that is print to device called jpg implicit no flux. Jpg okay so I am going to save that then go and we learn the code right so this time it was not explicit it was implicit right so let us run there is a problem okay. So let us go look at what the problem is it sees attempted to use the complex scale or as index okay assignment failed error called from in this code at line 46 columns 10 okay.

So this is the good thing about the error message is because when it gives error message it tells you clearly which line which column it has a problem so let us go look at.

(Refer Slide Time: 22:28)

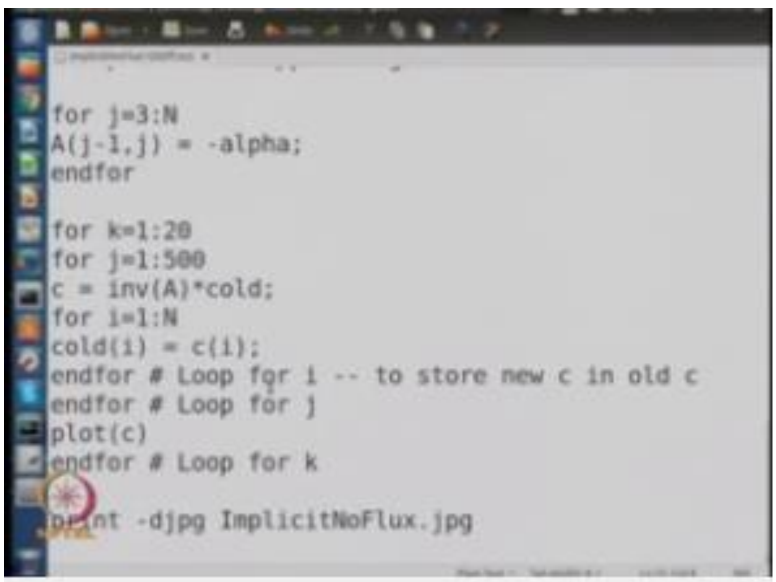


Line 46 you know line number and column number are given here at the bottom I hope you can see my moles so the line and column numbers are given here line 46 it says and I have made a mistake see this is J right it should have been small j it does not know what that J is and that is a reason why it gave the error result so as you can see and that is line 46 column 8 okay so it can give you quite accurate see.

It is a line 46 column 10 it was quite close to that point where the equal to side was that where it was complaining so one of the things that you have to learn when you are writing script or programming is that you should be able to read the error messages that you computer is giving you and understand what it is and go back and correct it. Okay in my definition a programmer is the one who can carry out a conversation with the computer you will do something computer will tell you something then you should be able to understand what it is and go corrected.

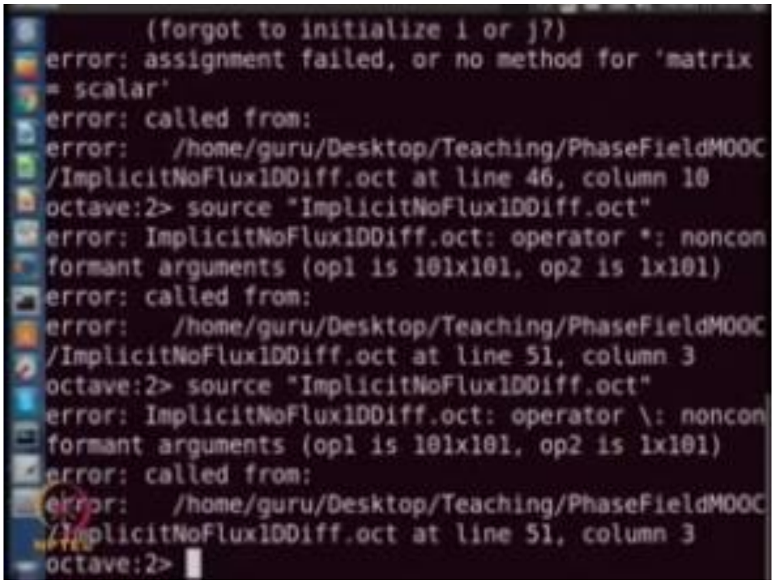
Typically first time programmers stent to you know copy paste the error message in Google and try to solve that way you will never become a good programmer okay you do not need and intermediary to talk when you have to talk directly with the computer yourself, okay so it is a good idea to read the error messages carefully try to understand yourself what it means and this is just practice every time get an error message go try to locate the error yourself it is like a puzzle okay so it say there is something wrong with that line what is wrong so you go figure it out and that is how your programming skills will improve.

(Refer Slide Time: 24:17)



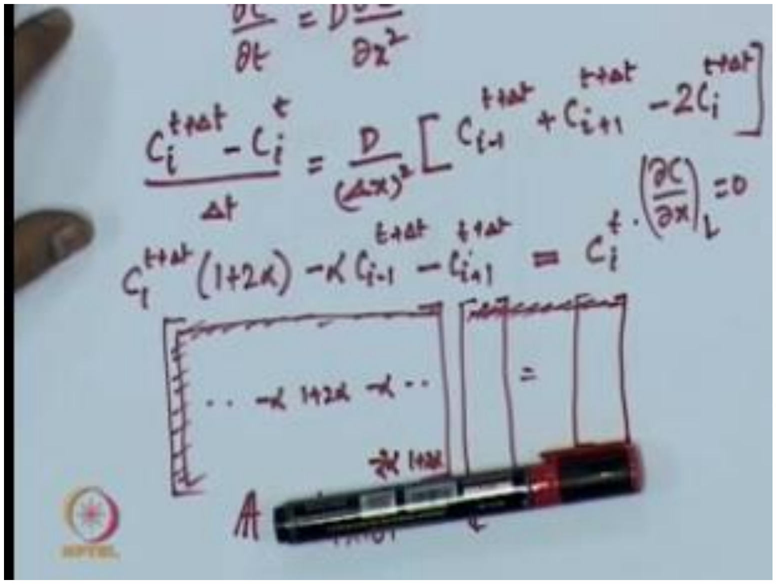
So let us try to run again okay so I still have a problem at line 51 columns 3 okay so let us go and look at what is wrong with line 51 columns 3, okay I do not understand what is the problem here okay so it says line 51 column 3 is a some error.

(Refer Slide Time: 24:49)



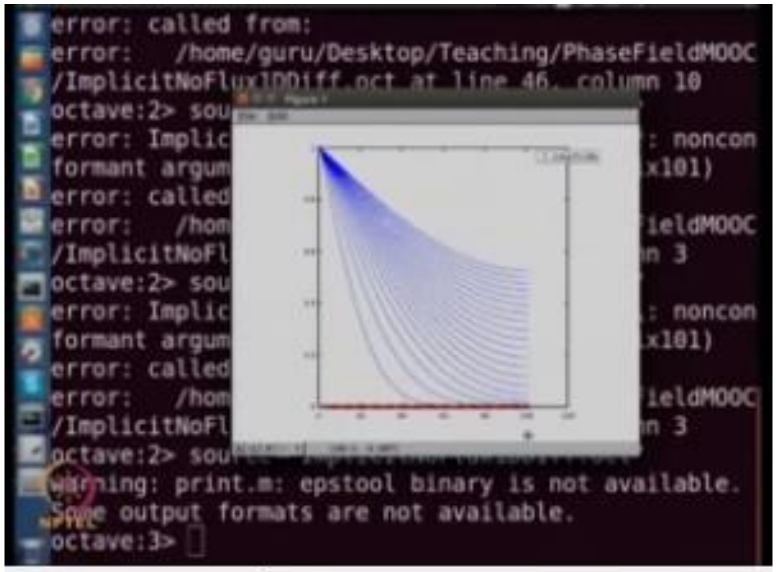
The operant one is 101/101 operant 2 should be 101/1 but it some of thing that it is 1/101 okay I do not understand why nit is so, so let us go back and try to do this other exercise let us say there is another way of doing the same thing c is a/hold let us see if this words let me see if this fixes my problem may it has yes that has okay, so if you are here.

(Refer Slide Time: 25:41)



You will hear my compute making lots of noise the reason is the explicit method is very very computationally simple the implicit method it is quite costly okay it has to do this inversion of a matrix and that is not easy okay.

(Refer Slide Time: 26:06)



So my computer is actually making lots of noise at this moment it is struggling but it got the solution and you can see the solution like in the earlier case so you have one and then you have this so the red point are basically the initial profiles is it is starts from here and it is like this and as time proceeds you are having this composition profile which is changing and like in the earlier case of course the boundary condition is the same whether it is solver it using implicit or explicit does not matter it is going to show you the same solution and of course it is showing with the same solution.

Okay now you can actually go and put ls so you can see that implicit no flux that Jpg is figure saved explicit not flux .Jpg is another figure saved so this figures are saved in the same directory in which you are going to invoke the scripts then the scripts is run the figures are generated a copy of it will be saved in the same direction for your future use. So we have done today using implicit and explicit key for one boundary condition how to solve the diffusion.

So in the next lecture we will go ahead and solve for let us say that at the left end also you have Dirichlet boundary condition solved okay so what happens so suppose I say that he left hand is always kept at some composition 0 it does not change so whatever material that comes then what will happen the material will escape from that point and the flux will be determine the how much material is coming in because the composition has to be at 0.

So whatever amount of material that is coming in should escape so that will be determine the flux so you can give Dirichlet boundary condition at they write explain and also and we will do that and we will solve again that using implicit and explicit scheme and that will be the topic for our next lecture. Thank you.

NPTEL
Principal Investigator
IIT Bombay

Prof. R.K Shevgaonkar

Head CDEEP

Prof. V.M Gadre

Producer

Arun Kalwankar

Digital Video Cameraman

&Graphics Designer

Amin B Shaikh

Online Editor

&Digital Video Editor

Tushar Deshpande

Jr. Technical Assistant

Vijay Kedare

Teaching Assistants

Arijit Roy

G Kamalakshi

Sr. Web Designer

Bharati Sakpal

Research Assistant

Riya Surange

Sr. Web Designer

Bharati M. Sarang

Web Designer

Nisha Thakur

Project Attendant

Ravi Paswan

Vinayak Raut

NATIONAL PROGRAMME ON TECHNOLOGY

ENHANCED LEARNING

(NPTEL)