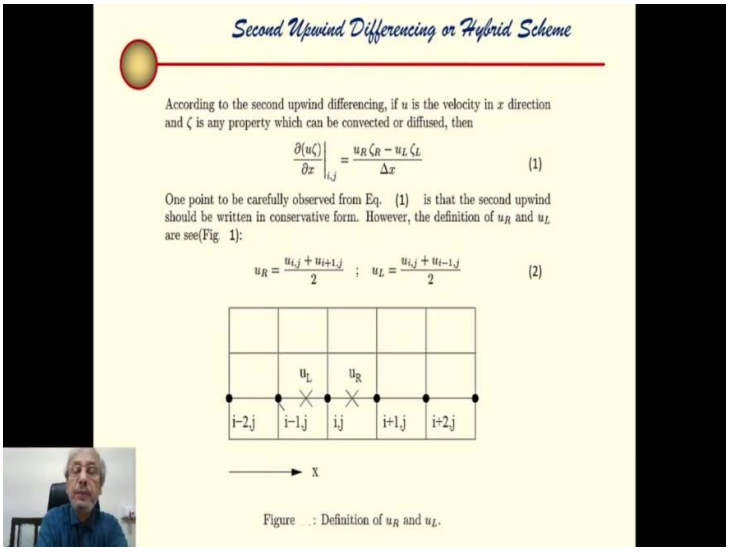
Computational Fluid Dynamics and Heat Transfer
Prof. Gautam Biswas
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur

Lecture – 07
Important Aspects of flow modeling – 3

Good afternoon. Today, we will discuss about some other important aspects of flow modeling. We have already discussed a few, some other aspects are to be covered today.

But before starting the actual lesson for today, we will recapitulate what we did in the last class; some important points that we did in the last class. As we mentioned in the last class that second upwind differencing is an improvement of first order upwind differencing and this is built-in with most of the popular codes that are in used such as Ansys fluent, CFX etc. So, it is, you know, of interest to know about it. We have already seen what it is; but even then, we'll just recapitulate again.

(Refer Slide Time: 00:35)

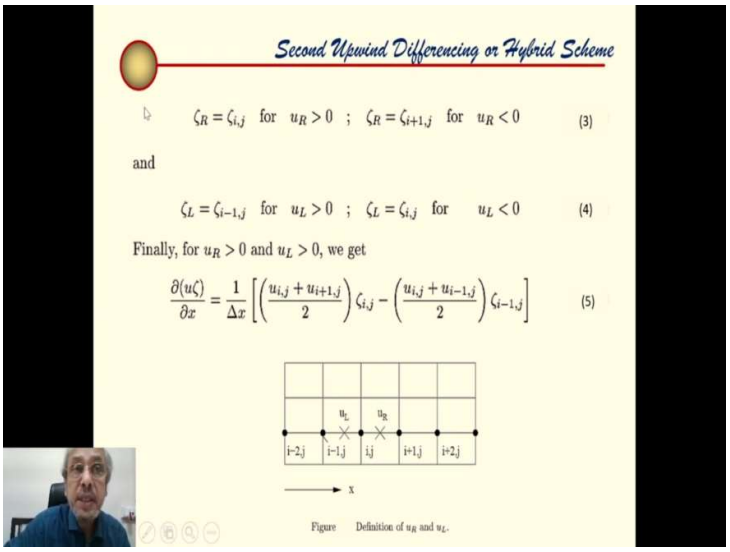


Let us look at one-dimensional space defined by i, j; i+1, j; i+2, j in one direction, another direction i-1, j; i-2, j and u_R is the point which is at the right-hand side of i, j; u_L is the point which is at the left-hand side of i, j. u_R is defined by equation 2; u_R and u_L definition, we can get here, this is exactly midway between $u_{i,j}$ and $u_{i+1,j}$ in the eastern direction or in the right direction.

And in the left direction or west direction, it is again exactly half between i-1, j and i, j and we are saying u_L is located here and defined here and u_R is defined here and u is the carrier velocity. It is transporting may be velocity itself, it may transport temperature, it may transport concentration of a species. So, basically u is a carrier velocity.

Now, also the basic principles of upwinding are- if u_R is positive, then ζ which is being transported, ζ_R will be at the upstream point. So, here if u_R is positive, then ζ will be defined at i, j- $\zeta_{i,j}$. If u_R is negative, then ζ will be defined at i+1, j.

(Refer Slide Time: 03:17)

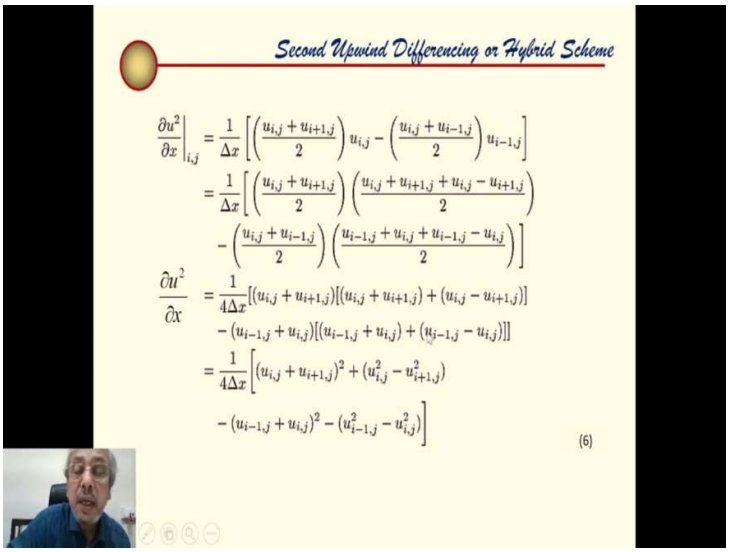


Similarly, if u_L is positive, then ζ_L will be defined at the upstream point which is $\zeta_{i-1,j}$ and if u_L is negative, then ζ will be defined at point i, j; rather downstream point, if u_L is negative. Rather I should say, then also it is upstream, because if u_L is negative, then i, j point is the upstream point with respect to negative velocity. And if u_L is positive, then $u_{i-1,j}$ is the upstream point.

So, here what we have done? We have this, we can see that $\frac{\partial(u\zeta)}{\partial x}$ which is $\frac{1}{\Delta x}$ and then, u_R we have associated $\zeta_{i,j}$ with that which is at the upstream point and u_L, we have associated $\zeta_{i-1,j}$ with that. This is also ζ at the upstream point.

So, we have taken u_R and u_L both are positive. We are doing our basic derivation based on that; but later, we will see that our derivation would be such that even if we change the direction of u_R and u_L, the formulation will remain valid.

(Refer Slide Time: 05:27)



So, here from the last expression that means, expression (5), what we have done? We have basically substituted this $\zeta_{i,j}$ by its, since it is derivative of u velocity; $u_{i,j}$, minus, again, u_L into $u_{i-1,j}$. Then, what we have done? We have basically added with this $u_{i,j}$; you can see in the bracket $u_{i+1,j}$ and subtracted that. And so, basically you have written $(u_{i,j} + u_{i+1,j} + u_{i,j} - u_{i+1,j})$.

So, if we really perform the algebra, this is nothing but $u_{i,j}$. $u_{i,j}$ we have written this way. Similarly, $u_{i-1,j}$, we have written as $\frac{u_{i-1,j}+u_{i,j}+u_{i-1,j}-u_{i,j}}{2}$. And again, if we perform this algebra, we will get back $u_{i-1,j}$. This is a sort of you know rearrangement.

Then, we have you know this 2 into 2 at the denominator, we have transferred to the outside the bracket $\frac{1}{4\Delta x}$ and then, we have tried to regroup the basically different bracketed terms. Here, it is $u_{i,j} + u_{i+1,j}$. We have multiplied it with this, because it is multiplying this entire quantity, we have just grouped this. We have $u_{i,j} + u_{i+1,j} + u_{i,j} - u_{i+1,j}$.

The second, I mean, term that means, associated with u_L. So, here also $u_{i,j} + u_{i-1,j}$, this we have first you know we are multiplying with the entire quantity defined at the numerator, but we have written $u_{i-1,j} + u_{i,j} + u_{i-1,j} - u_{i,j}$.

Idea is that, see here we are getting basically $(u_{i,j} + u_{i+1,j})^2$, you can square first multiplication. Second multiplication is given you know it is like $(a + b)(a - b)$, so, $(a^2 - b^2)$. Then, third multiplication again $(u_{i-1,j} + u_{i,j})^2$. Again, the fourth term is $(a + b)$ being multiplied with $(a - b)$. So, $(u_{i-1,j}^2 - u_{i,j}^2)$ and this way, writing this has some advantage. We'll be able to appreciate in the next slide. What we have done that we have in the next slide; we have retained this term.

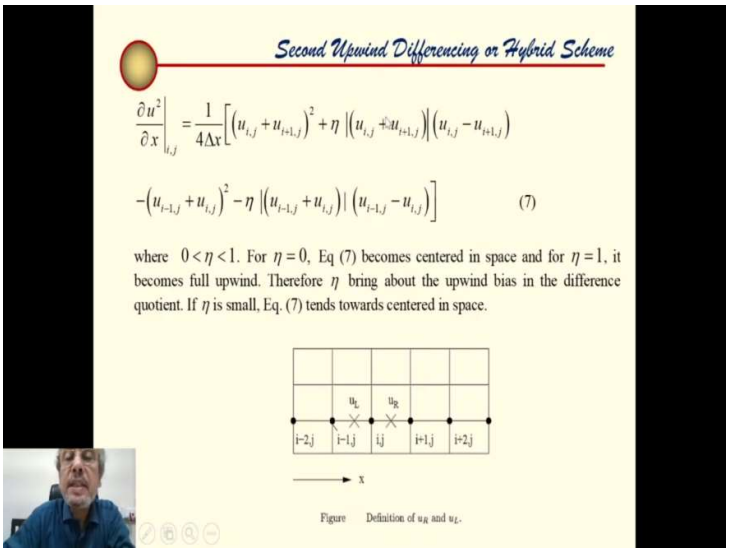
But this term, we are writing as $(u_{i,j} + u_{i+1,j})(u_{i,j} - u_{i+1,j})$. Again, $(a^2 - b^2)$ is being again split as $(a + b)(a - b)$. But while doing this we are placing a modulus sign here. So, this is modulus of this quantity.

Similarly, if you go to the previous slide, we'll retain this term of equation 6; but this term $(a^2 - b^2)$, we'll again split into $(a + b)(a - b)$, but this time this term $(a + b)$ is under a modulus sign. Now, this under a modulus sign, we have placed strategically because this is basically defined at the right nodal point velocity u and again, at the left nodal point velocity L.

This quantity $(u_{i-1,j} + u_{i,j})$ at the u_i, this is basically at the location L, location u_L. Now, if u_L is positive, based on that we derived it, it is valid; but if u_L and u_R, if these are negative, then also you will see this remains valid. Because even if it is negative, it will assume the modulus sign and calculation will give it same.

So, you now, we are introducing this modulus sign in order to have locally u_R and u_L be flexible. This can be positive as well as negative, u_L can be positive as well as negative. And then, we have also introduced one symbol η. η is basically between we have written here; η varies between 0 and 1.

(Refer Slide Time: 09:03)



If you place η equal to 0, you can see this whole differencing $\frac{\partial u^2}{\partial x}$ will be u at R point squared minus u at L point squared. So, this is basically central differencing. And if η is 1, then you will see this will be basically first order upwind differencing.

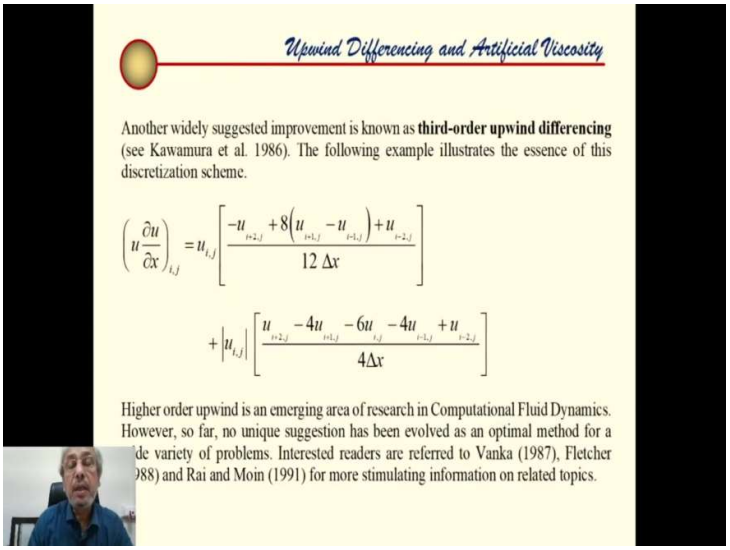
So, we will vary η between 0 and 1; preferably η should be kept as low as possible. Why I am saying as low as possible because presence of η is needed. This will give the upwind bias. Upwind bias is needed for retaining transportive property. And making the

formulation such that it never violates transportive property. But if upwind is done in such a way that it increases artificial viscosity, then we are introducing some error.

We have learned about artificial viscosity, relationship between upwinding and artificial viscosity. So, more contribution from upwinding will increase the artificial viscosity more, that is why η is needed. But it should be as minimum as possible. It should just maintain; it should be good enough to maintain transportive property.

But at any point of time, it should not contribute to large artificial viscosity or significant amount of false diffusion. So, usually, you know, while working with a scheme, some trial run is given; but it is preferable to retain this η value somewhere near 0.2; maybe you know 0.23, 0.24 etc. It is tested that serves a purpose.

(Refer Slide Time: 14:05)



Now, we will, based on this, we will start our today's discussion on even higher order upwinding scheme. So, this is a very well-known higher order upwinding scheme, which is known as third-order upwind differencing of we have written here, the name of the first author Kawamura, second author is Kuwahara, there is a third author also Takami. So, but all of them worked with Professor Kuwahara. Professor Kuwahara at that time, he was in Tokyo Institute of Technology, He's a very well-known fluid dynamicist and this scheme was devised by him. He devised this scheme to solve the turbulent flow problem, as such you know, he attempted direct numerical simulation using this scheme so that false diffusion is almost neglected or zero and other form of discretization errors are also minimized. You can see this is third order upwind differencing.

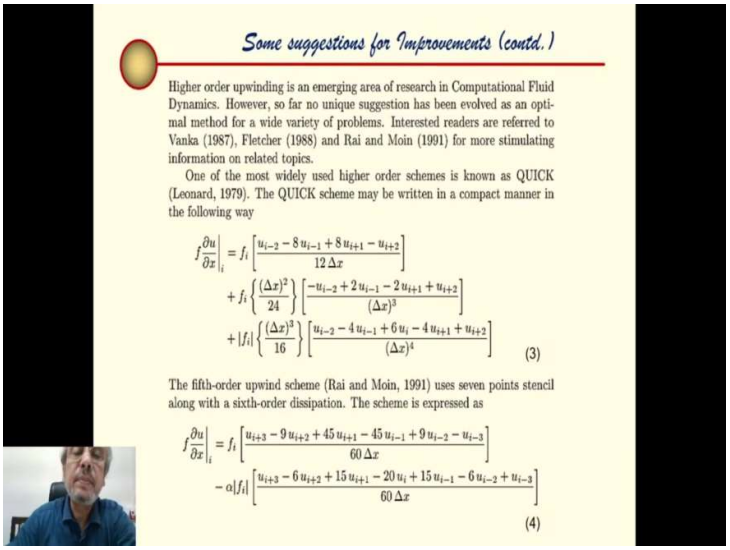
So, basically second order error which gives the first leading term of artificial viscosity, $\nu_{artificial}\frac{\partial^2 u}{\partial x^2}$ that $\nu_{artificial}$ will not be generated. It will have even higher order errors, maybe fourth order, fifth order errors, which are, you know, basically dispersion error, phase error, those errors may be introduced.

But if the final grid is taken, those error also will be minimized; but second order error is not there, this is third order accurate scheme. And I have written it only in one-dimension. So, you can see how it has been written and it has involved i+2, i+1, i-1, i-2, and obviously i, j point.

So, i, j point and on either side two points, so this is called five-point stencil. So, in order to find out derivative at a point, you need, in one direction, you need two points in the downstream direction, two points in the upstream direction and it is called five-point stencil. And here also, you can see this $u_{i,j}$ is placed under modulus sign. So, that it can handle positive u or negative u at the point of interest at the i, j point. The expression will eventually be valid.

Now, again, let me remind that this has been done in one direction and the model discretization on one-dimensional paradigm. In three-dimensional paradigm, these terms will be little more complex because no quantity is one-dimensional quantity in three-dimensional paradigm. So, very interesting discussions are available. I have given some references; Professor Vanka's reference, Professor Fletcher's reference and another reference Rai and Moin. We will discuss about that scheme also in the next slide.

(Refer Slide Time: 18:03)



So, next slide since we are now discussing higher order discretization schemes, we will discuss a very popular scheme which was developed by Professor Leonard in 1979. This is called QUICK scheme. QUICK is acronym, it means Quadratic Upwind Interpolation of Convective Kinematics. And you can see here also QUICK scheme involves point of interest, that means, i and then, the upstream points i-1, i-2; downstream points i+1, i+2. So, this is also based on five-point stencil and here also you can see, here this f_i has been put under modulus sign, in order to take care of the sign of f. We have given deliberately f, so it was written that way.

So, f is basically the velocity of interest. It is u_i in our case and sign of u_i, whether it is positive or negative, we'll place it here and it will eventually take the correct form. This scheme, QUICK scheme is a very popular scheme and this is also available in most of the commercial codes. Next scheme what I will mention, this is not available in any of the commercial codes. This is again developed by Stanford Group or many of you might be knowing, a very famous place in Stanford University called Center for Turbulence Research, there Professor Parviz Moin and M. M. Rai, they developed this scheme in 1991 and this is for doing Direct Numerical Simulation (DNS).

Now, a side remarks is that although, Professor Kuwahara also developed this scheme for doing direct numerical simulation. QUICK was also developed with the basic target of doing direct numerical simulation. Direct numerical simulation, we'll discuss later in little more detail, where you do not use any model for turbulence modeling. You take very very fine grid and you basically use the numerical scheme which produces absolutely no error and the grid size is such that it is below the smallest fluctuation, smallest turbulence fluctuation which is known as Kolmogorov scale. So, if you want to go for such fine grid size maybe for a given domain, you may require 10^{12}-10^{14} grid points and obviously, in three-dimension with so many variables and with 10^{12}-10^{14} grids, you cannot solve using ordinary computers.

The program has to be parallel program and it has to run on basically super computers. High performance computing platform is needed for that. So, target was to address DNS. But in practice today, I mean again, this is a passive remark. This Professor Kuwahara's third order upwinding or QUICK scheme, these are used for very very accurate simulations; but not for DNS. They are used for another very accurate turbulent flow simulation technique called LES, Large Eddy Simulation. And you know several other intricate flow problems, where very high level of accuracy of discretization is needed. These 2 schemes QUICK and Professor Kuwahara's scheme is called third order Kuwahara's scheme. These are both used for all such purposes. But for DNS, its acceptability is not that much. We do not have the right time to discuss all those issues. I will just mention the what happens in practice.

Now, I will remark on Rai and Moin scheme. Rai and Moin scheme is a scheme which is used for DNS calculation. Very popular scheme and basically another such scheme is called Lele's Compact scheme that is also very accurate used for the purpose of DNS. And was developed by Professor Sanjiva Lele in again Stanford and this is Rai and Moin scheme, we are mentioning here. Here also you can see that it is fifth order accurate. Formally, QUICK and Kuwahara's third order scheme, name itself is saying these two schemes are formally third order accurate, but Rai and Moin scheme is fifth order accurate. And it involves you can see seven points; i is the point of interest, these are downstream points, i+1, i+2 and i+3, these are downstream points and these are upstream points i-1, i-2 and i-3. So, i-1, i-2, i-3; three upstream points, three downstream points and the point of interest, seven points are needed for this accurate discretization and this is called seven-point stencil. This f_i again, this is a carrier phase velocity. You have seen that we have put it under modulus sign and that means, it can handle whether locally the velocity is positive

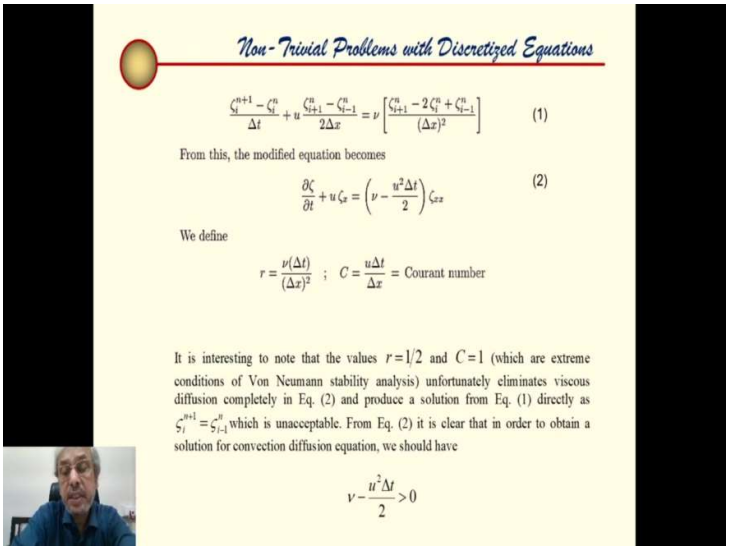
or negative that doesn't matter. This alpha is a factor it can be varied; usually it is 1, but it can be made you know more than 1 also. It takes slightly different form, but usually alpha equal to 1 works well.

Now, I will wrap up this accuracy related discussions. We have only discussed about formal accuracy of convective terms; specifically, one convective term, this way other convective terms can be discretized following higher order discretization method. But accuracy means something else, it is a combined aspect.

So, it is accuracy of convective terms, accuracy of discretization of diffusive term, then accuracy of grid related error; whether grid is fine enough, whether grid is skewed or you know grid related errors are in dominating and then, temporal discretization or I mean usual temporal discretization, we have mentioned here is a first order discretization whether u_i or $u_{i,j}$ or $u_{i,j,k}$ at $n+1^{th}$ level minus u_i or $u_{i,j}$ or $u_{i,j,k}$ at n^{th} level by Δt. So, we have seen also in last lecture that from the truncated terms of discretization error, second order error can come. So, therefore, there are ways to have higher order temporal discretization these are called Adam-Bashforth scheme. I mean Runge-Kutta schemes are also used for progressing in the time direction and maintaining higher order accuracy.

But all these are basically needed, such a high level of accuracy is basically needed, where flow with very high level of accuracy as I said that the smallest fluctuation of turbulence or similar such accuracy is needed. Usually, in engineering flows such high-level accuracy may not be needed, but it should be reasonably accurate. So, you will see in most of the problems, we will try to go for higher order accurate scheme; preferably a second order discretization scheme of the convective term and diffusive terms are usually second order discretization method is followed for diffusive terms. And the temporal discretization in some cases, we go for second order accurate by adopting Adam-Bashforth method.

(Refer Slide Time: 29:39)



And we also pay attention to the implementation of the boundary conditions very accurately. Having done that, we will just try to wrap up finite difference-based nuances of discretization technique because these are some of the aspects one should know before handling a flow and heat transfer problem. Now, we will just touch upon a problem, we have given a heading- Non-Trivial Problems with Discretized Equations.

Now, you can see equation 1 is Burger's equation. We have discretized the temporal term you know forward in time, first order forward in time. The convective term $u\frac{\partial u}{\partial x}$, we have discretized using central difference and diffusive term is obviously, we have discretized using central difference. And then, we have, see if ζ_i^{n+1} level written in Taylor series in time direction. Whatever expression we'll get from there $(\zeta_i^{n+1} - \zeta_i^n)/\Delta t$, we'll try to regroup and see the remaining terms. Similarly, we'll perform this exercise for the convective term and the diffusive term and if you recall in last class, we did we'll try to reconstruct the PDE and then, here as you can see, we have gotten back this PDE.

But this PDE instead of having $\nu\frac{\partial^2 \zeta}{\partial x^2}$ or $(\nu\zeta_{xx})$, it has $\left(\nu - \frac{u^2\Delta t}{2}\right)\zeta_{xx}$ and this error is primarily coming from the temporal discretization scheme. Also, we have explained in the last class, how the truncated terms of the temporal discretization can be expressed in terms of spatial discretization, the spatial variables.

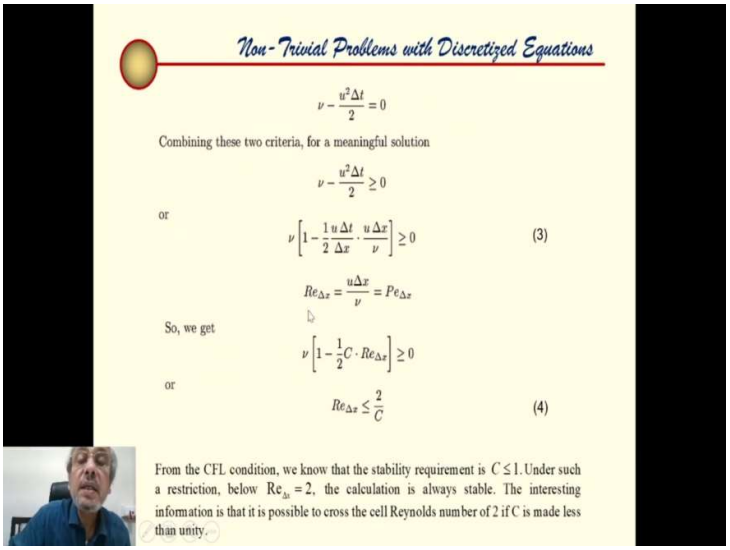
So, it can be written this way. Now, r is a quantity, we can define $\frac{\nu(\Delta t)}{(\Delta x)^2}$. C is $\frac{u(\Delta t)}{(\Delta x)}$, it is Courant number and this is called actually Grid Fourier number. Although, we have not coined this term so far. Now, for r equal to 1/2 and C equal to 1, we can see this equation even though we are attempting to solve this equation. This equation will be completely able to dispense with the viscous diffusion term completely. This will become an equation similar to Euler's equation, it is $\frac{\partial \zeta}{\partial t} + u\zeta_x = 0$. Second thing is that if we substitute r equal to 1/2 and C equal to 1 at equation (1), equation (1) let us multiply by Δt throughout; let us divide by you know Δx and then, we'll be able to get the you know I will try to write what is ζ_i^{n+1}.

After doing that, we will see straight away, we get $\zeta_i^{n+1} = \zeta_{i-1}^n$; straight away we'll get the solution ζ_i^{n+1}. Just do this faithfully, you know transfer, keep the temporal term on the left-hand side, transfer temporal derivative, transfer all other terms on the right-hand side. Then, multiply whole equation entire equation by Δt and then, divide basically the right-hand side by $(\Delta x)^2$.

You will see that right hand side; sorry, the term which came due to convective terms you divide by Δx which arises out of diffusive term you divide by $(\Delta x)^2$, we have written here. So, basically you transfer except for this temporal term; all other terms on the right-hand side and multiply use by Δx. You will get basically $\zeta_i^{n+1} = \zeta_{i-1}^n$. So, your solution is obtained. But this is not the solution, which is not acceptable. So, in order to get physically

meaningful solution, $\left(\nu - \frac{u^2\Delta t}{2}\right)$, this has to be greater than 0. This has to be a nonzero term.

(Refer Slide Time: 35:19)



And 0 term is extreme case, if we get 0 term from there, it can be in the Euler's equation. So, combination is $\left(\nu - \frac{u^2\Delta t}{2}\right)$ must be greater than equal to 0. Now, we can sort of reposition regroup this $\frac{u^2\Delta t}{2}$ as $\nu\left[\frac{1}{2}\frac{u\Delta t}{\Delta x}\cdot\frac{u\Delta x}{\nu}\right]$.

Then, you know this is basically $\frac{u\Delta t}{\Delta x}$ nothing but Courant number (C); $\frac{u\Delta x}{\nu}$ is called grid Peclet number ($Pe_{\Delta x}$) or grid Reynolds number ($Re_{\Delta x}$). Different books use different terminology; some books they call it grid Peclet number, grid Peclet number, some books it is grid Reynolds number, whatever it is basically it is representative of Reynolds number based on the dimension of the grid.

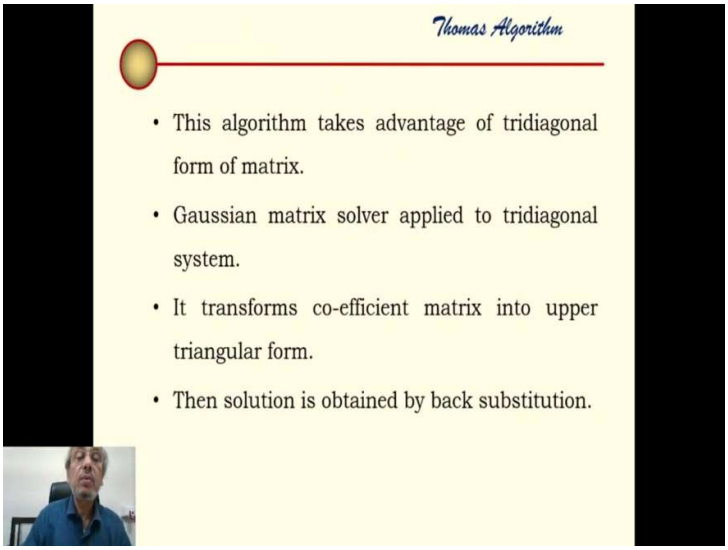
So, from this identity, we can see this equation that $\nu\left[1 - \frac{1}{2}C\cdot Re_{\Delta x}\right] \geq 0$; ν is non-zero. From here, we will we get $Re_{\Delta x} \leq \frac{2}{C}$. Now, in many places, it is loosely told that grid Reynolds number or grid Peclet number should be less than 2 which is we are also not violating that, we are saying it should be less than 2.

So, upper limit is 2. But actually, it is less than equal to 2 by C, 2 by Courant number, then depending on the value of the Courant number, this grid Peclet number, we can have a little more affordable value because grid Peclet number is basically you know we will define your grid density. Here it is in x direction, similarly it can be in y direction, it can be in z direction or in (r, θ, z) coordinate, it will be basically in θ-direction or in r-direction.

So, whatever is the direction in that direction depending on the grid dimension, this is basically grid Reynold's number or grid Peclet number. Grid density will dictate what is the value of the grid Peclet number or grid Reynold's number in the domain and I said, for getting a stable meaningful calculation, the restriction is grid Reynold's number should be less than 2.

But we have just modified it actually grid Reynolds number is less than 2 by C and since C is always less than 1, we can you know get this you know upper limit of grid Reynolds number more than 2. So, it can exceed 2, but not more than 2 by C. So, that is what we have written that under such it is restriction below grid Reynolds number 2 calculation is always stable. The interesting information is that it is possible to cross the cell Reynolds number or grid Reynolds number of 2, if C is meant less than unity; C is the courant number.

(Refer Slide Time: 39:31)



Now, so far, we have been discussing about tridiagonal matrices. We have seen always our tendency was to formulate in such a way so that we can get tridiagonal matrix. Like in two-dimensional problems, we discussed, we mentioned about ADI scheme means Alternating Direction Implicit scheme, in each alternating direction, we'll get tridiagonal matrices and why we are always trying to do that? Because it is very easy to solve tridiagonal matrix.

If we get non tridiagonal matrix, you know the solution is little more involved. As such there is another similar such matrix which is called you know Penta-diagonal matrix. There are certain algorithms which can solve also Penta-diagonal matrix; I mean not as easily as tridiagonal matrix, but though I mean there are special techniques to solve Penta-diagonal matrix also.

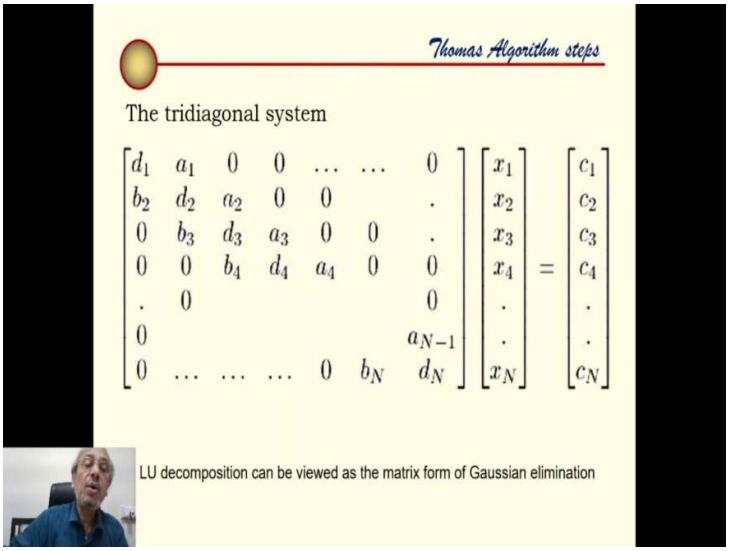
Now, tridiagonal matrix algorithm is called Thomas algorithm, named after Thomas, who conceptualized it. Because it's basically seeing something much beyond. Looking at a

tridiagonal matrix, it was not otherwise possible to know that solution technique is so simple.

So, this algorithm takes advantage of tridiagonal Thomas algorithm. This takes advantage of tridiagonal matrix system. Gaussian matrix solver is applied to tridiagonal. Actually, Gaussian-elimination method which gives us of any arbitrary square matrix gives us upper triangular matrix and then, solution becomes simple.

Similar operations are done you know as such on tridiagonal matrix also. An algorithm is set by summarizing those operations, but final algorithm when we get, it is very easy. So, we transform the coefficient matrix into upper triangular form, then the solution is obtained by back substitution.

(Refer Slide Time: 42:33)



So, let us highlight the nomenclature. After getting the tridiagonal matrix, the tridiagonal system, coefficient matrix is given here. In the coefficient matrix you can see, we have not used i, j indices. We have only used you know 1 index. Because we will just identify 3 diagonals; principal diagonal, you can see.

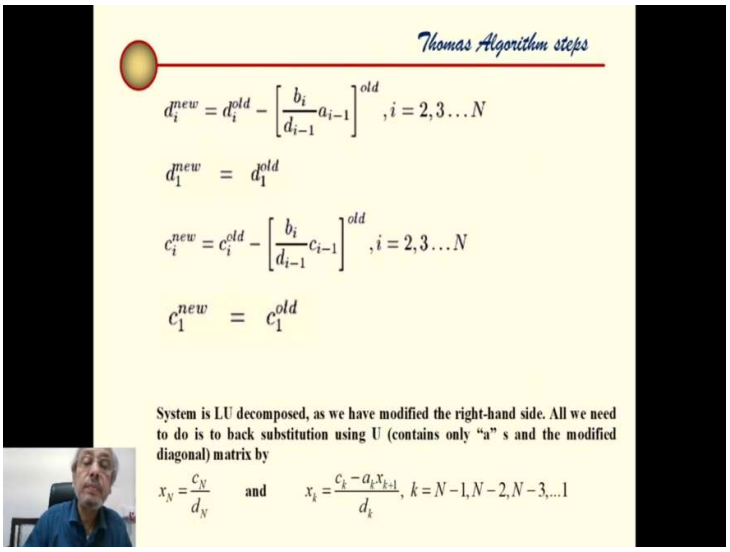
He is having n number of elements; d_1, d_2, d_3, d_4 up to d_N and immediate the diagonal below it, you can call it sub-diagonal b_2, b_3, b_4; one element will be less counted. So, it is up to b_N, b_1 is not there. So, b_2, b_3, b_4 up to b_N. Again, N number of elements, one is less counted. So, one less is counted. So, basically (N-1) elements.

Similarly, a_1, a_2, a_3, a_4 which is the diagonal at the above the principal diagonal, we can call it super-diagonal also; 1 2 3 4 up to a_{N-1}. So, the elements should have an array with dimension (N-1); d elements should have an array with dimension N; b elements should have an array with dimension again (N-1).

And x_1, x_2, x_3, x_4 to x_N, this x vector column vector these are all unknown elements and right-hand side c_1, c_2, c_3, c_4 up to c_N again a column vector and this of N number of elements are there, this is also known. So, x_1 to x_N are unknown, all c's are known, all d's are known, all a's are known, all b's are known. We will find out x_1 to x_N.

Now, it is this operation as I said, it will be Gaussian-elimination kind of operation. But you know it will finally, transform coefficient matrix into an upper triangular matrix. So, this is in a way, we can call it LU decomposition.

(Refer Slide Time: 46:05)



So, LU decomposition can be viewed as the matrix form of Gaussian elimination. And see in all these, what we will store is basically this d array, this d, I mean, elements, a changed d array, changed c array, nothing else will change.

But as I said behind all Gaussian elimination operations are on to convert the matrix into upper triangular matrix, but in order to know x_1 to x_N, we will just need diagonal elements of that upper triangular matrix and this c_1 to c_N this array. So, changed array c_1 to c_N, this has to be stored and changed array d_1 to d_N, these have to be stored; other arrays will also be needed for starting the calculation.

So, we will start the calculation with four arrays; d_1 to d_N, a_1 to a_{N-1} and b_2 to b_N and the fourth array is c_1 to c_N. And now, we will change with the help of the mathematical operation and b and a array, d array and similarly, with the help of b and a array and the diagonal elements we'll change c array. But storage of changed array when we'll consider? Changed array will have only changed d array and changed c array. From, there we will be able to get all the values of x_1 to x_N.

Now, first we will show how this d array will change from d_1 to d_N. Then, we will show how c_1 to c_N will change. So, you can see d_1^{new} is d_1^{old}. So, first element of d array will not change, this will remain d_1^{old}; but all other elements d_2 to d_N will change.

This is

$$d_i^{new} = d_i^{old} - \left[\frac{b_i}{d_{i-1}} a_{i-1}\right]^{old}$$

i is varied from 2 to N. So,

$$d_2^{new} = d_2^{old} - \left[\frac{b_2}{d_1} a_1\right]^{old}$$

So, this will be this minus, then old value of d_2 minus again b_1. Now, it will be needed, basically b_2 divided by d_1 into a_1. Similarly,

$$d_3^{new} = d_3^{old} - \left[\frac{b_3}{d_2} a_2\right]^{old}$$

So, this way entire array d_1 to d_N will be changed. Now, c array will also change.

$c_i^{new} = c_i^{old} - \left[\frac{b_i}{d_{i-1}} c_{i-1}\right]^{old}$;

$c_1^{new} = c_1^{old}$sorry. So, first value will not change, $c_1^{new} = c_1^{old}$. Now, from c_2, so $c_i^{new} = c_i^{old} - \left[\frac{b_i}{d_{i-1}} c_{i-1}\right]^{old}$; that means, i = 2 means $c_2^{new} = c_2^{old} - \left[\frac{b_2}{d_1} c_1\right]^{old}$

So, this way, we will change all the elements here and these two changed d and changed c are stored and then, we will get back the values of x_1 to x_N. The system is LU decomposed. As we have modified, the right-hand side all we need to do is to back substitution using U matrix which is a upper triangular matrix now. This contains only "a" s; "a" s are there, "a" s we have not changed and the modified diagonal.

So, now, the new upper triangular matrix effectively after doing Gaussian elimination operation, the new upper triangular matrix will have old a's and new d's and new c's. So, the last term, we can well imagine that it will be x_N and if these are all 0. So, changed d_N and changed c_N. So, x_N will be that changed new c_N by d_N and then, we can set

$$x_k = \frac{c_k - a_k x_{k+1}}{d_k}$$

and k will vary from (N-1), this is back substitution.

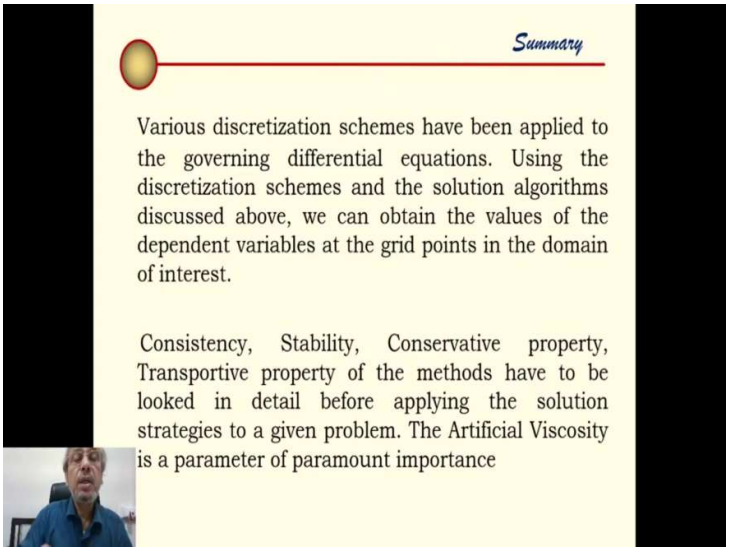
So, first term is x_N, second term is $x_{N-1} = \frac{c_{N-1} - a_{N-1} x_N}{d_{N-1}}$; x_N has been just found out, because now you know just d_{N-1} or x_{N-1}, c_{N-1}, these terms are involved. So, we will find out x_{N-1} from this expression. Then, x_{N-2}, x_{N-3} up to x_1.

So, x_1 to x_N all the terms will be evaluated. So, and now all of us can appreciate why we go always after tridiagonal matrix algorithm. Because the entire system of tridiagonal matrix, where the coefficient matrix is having only the diagonal element and one diagonal below the diagonal element, another diagonal above the diagonal element, all other elements are 0; then, the x vector which is unknown vector equal to the c vector which is known vector.

And when we change this the new elements, all c new elements are found out, all d diagonal elements are found out and immediate super-diagonal elements like the elements above the diagonal a's are not changed at all and need not be stored because this was stored earlier.

So, only d and c's are new values, a new array has to be there to store these values and from there, making use of old a value, this new d values and new c values we'll be able to find out all unknown quantities.

(Refer Slide Time: 55:23)

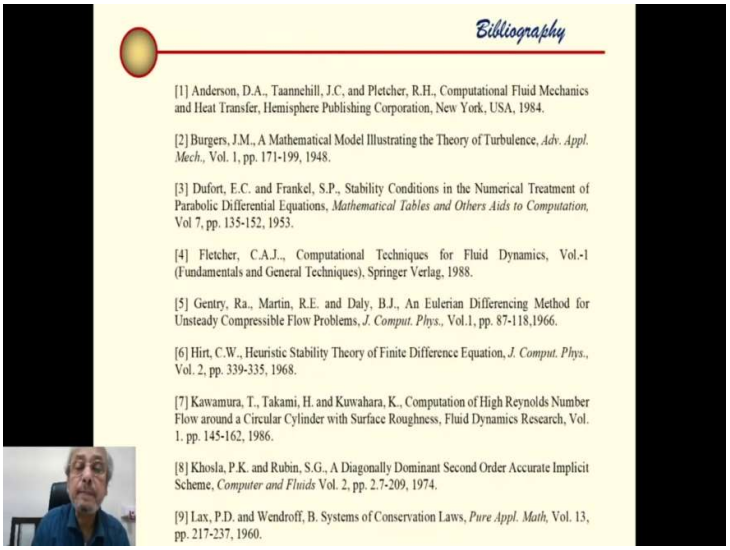


So, we have discussed various discretization schemes, applied to governing differential equations. Using discretization schemes and the solution algorithms discussed above, we can obtain the values of the dependent variables at the grid points in the domain of interest. That is what is our target. We have made ourselves acquainted with this knowledge, how to do that from fundamentals and we'll apply that for more complex problems.

Now, consistency, stability, conservative property, transportive property of the methods have to be looked into and have to be looked in detail, before applying the solution strategies to a given problem. Obviously, we have seen these are very important issues, otherwise by violating these issues, we may not get the appropriate solution. Even if we get the solution, results are spurious; many a times, the program will not work, it will break.

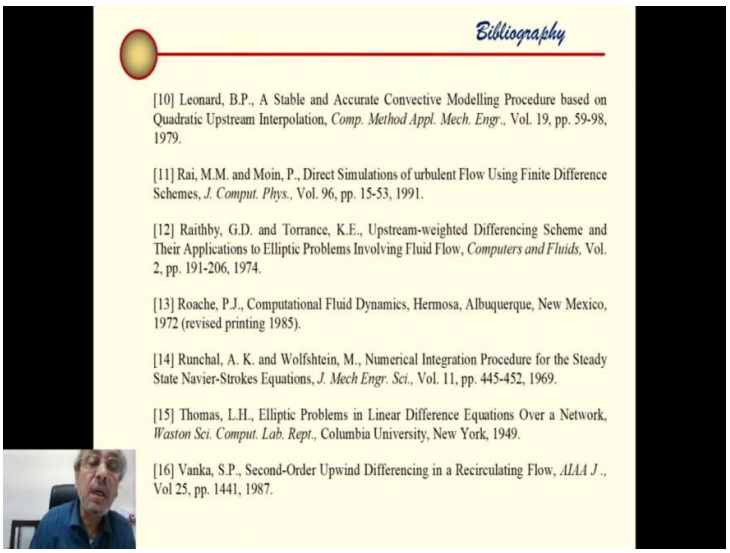
So, consistency, stability, conservative property, transportive property, these are to be really handled with care. Artificial viscosity is a parameter of paramount interest. It has to be there so that the algorithm works; but it should be minimized. So, that it is you know not a disturbance to the numerical values of the dependent variable or its distribution of the dependent variables in the domain.

(Refer Slide Time: 57:25)



So, while discussing our lessons, we have taken help from several literature, open literature which are available. I have just given a list so that when you prepare your lessons, you can always refer to these papers. I have mentioned it here.

(Refer Slide Time: 58:05)



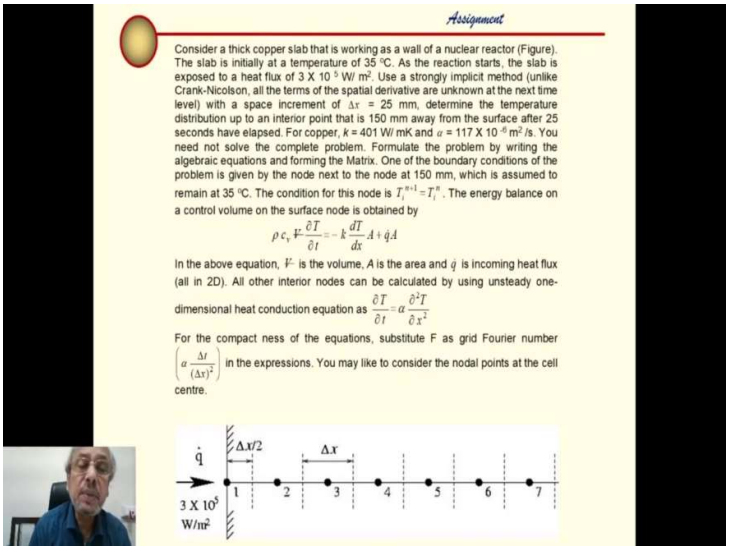
You can if needed, you can download it from the respective journals on the net. All these journals are subscribed by IIT Kanpur. So, it will not be a problem. Another point while discussing about algorithms, our assignments numerical assignments, finally, you have to solve the problems using computer.

So, you have to write your own codes and my suggestion is that whatever language you know, you use that language. If somebody knows maybe Fortran 77, somebody knows Fortran 90, somebody knows Fortran 95 all these are scientific languages, you can use. Somebody knows C, somebody knows C++ well. So, please use C, C++.

Somebody you know maybe you know enthusiastic about Python, so you can use Python. So, any language you know, please feel free to use the language. Only thing depending on the computer you are using; you check it whether your computer is having that compiler.

If your computer is endowed with you know C, C++, Python, FORTRAN, all these compilers, then use any language you like. Because when I will look at the solutions that you have done, I will you know suggest you to run the codes and the code should run and at least produce some sample results.

(Refer Slide Time: 60:11)



Now, we are given one example problem. This example problem, we will discuss very quickly how to really formulate a given problem. So, basically you can read it slowly. I will try to read it little fast that this is a slab and we are solving a one-dimensional problem. This is the flow direction x-direction; not flow direction, x-direction.

And we are solving a problem which is dependent on x because z-direction which is perpendicular to the plane of interest is infinite and problem is not changing in that direction. Also, y-direction which is perpendicular to x-direction, in that direction also problem description is not changing. So, the way basically temperature, this is a problem.

So, this is a slab. This physical domain is a slab which is you know which has very large z-dimension that is dimension perpendicular to the plane. And again, which is again you know very large dimension perpendicular to x line. So, whatever variation is observable that is changing only in x-direction, it is repeating in all y's, it is repeating in all z's.

That is why this is one-dimensional problem. So, this is a slab of a nuclear reactor and at the left-hand side, you can see some heat flux is coming and which is quite high $3x10^5$ W/m^2. This slab is exposed to that amount of thermal energy. So, obviously, thermal energy and this slab is a metallic slab made of copper. So, this slab will be heated and some time has been given.

After that time instant, you can see that, consider a thick copper slab that is working as a wall of a nuclear reactor. The slab is initially at a temperature 35 °C. As the reaction starts, slab is exposed to a heat flux of $3x10^5$ W/m^2. Using a strong implicit method, we have learned Crank-Nicolson implicit method.

Here, in the example problem, we'll do another variant also implicit, but strongly implicit method and with a space increment of $\Delta x = 25$ millimeters. Determine the temperature distribution up to the interior point which is 150 millimeters away from the surface after 25 seconds.

So, at initially the entire slab is having 35 °C temperature centigrade and it is exposed to big thermal flux. And then, after basically 25 seconds, we have to find out temperature distribution in the x-direction and that temperature distribution is valid in all y, all z.

So, formulate the problem by writing algebraic equations and forming the matrix. One of the boundary conditions of the problem is given by the node next to the node 150 millimeters. So, each node is separated by 25 millimeters. So, the last node is basically what is being said that after 25 seconds, last node temperature will not change.

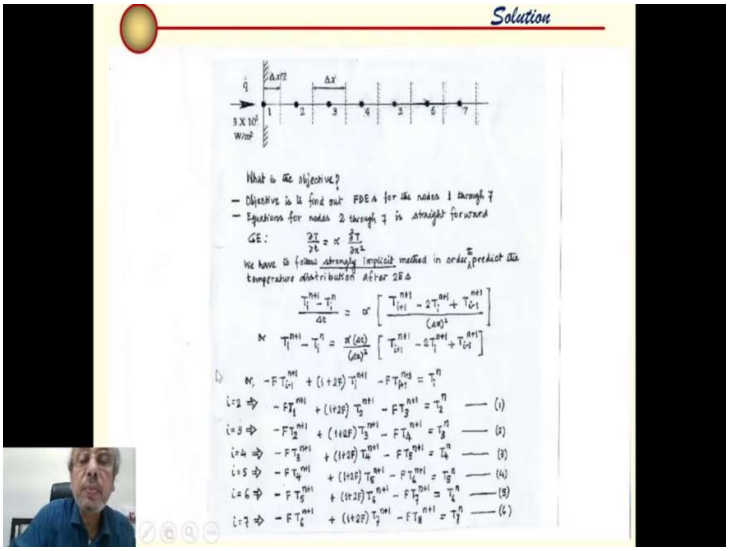
One of the boundary conditions of problems given by node next to 150 millimeters. So, this is 25, 25, 50, 25, 75, 25, 100, 25, 125, 25, again 150; maybe equal 25 millimeter away had there been a node 8, that temperature at that eighth node is not changing.

So, that is after 25 seconds that is one boundary condition known to you and also, at node number 1, the basically boundary condition is not known, but that has to be found out from this energy balance equation. This is basically accumulation of energy within this control volume. If we take you know a small control volume, comprising of $\frac{\Delta x}{2}$ in x-direction and maybe unit dimension is y and unit dimension in z-direction.

Then, that is the thermal energy and this is the thermal energy going out and this is the thermal energy coming in. So, in above equation $\forbidden{V}$ is, V with a cut mark, this is the volume. A is the area and $\dot{q}$ is the incoming heat flux, all other interior nodes can be calculated by using unsteady one-dimensional heat conduction equation.

For the compactness of the equation substitute a F as the grid Fourier number, compactness- there should not be any gap. This is somehow you know typographical mistake. So, $\alpha \frac{\Delta t}{(\Delta x)^2}$ wherever we will get, we will substitute that by F. Now, let us formulate the problem.

(Refer Slide Time: 67:09)



So, basically this is the governing equation, the one-dimensional heat conduction equation;

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

and for see first node is boundary condition, we will determine it. Last node is also boundary condition, we will determine it separately.

But you know all intermediate nodes, this equation is applicable, where we can write

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \left[\frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{(\Delta x)^2} \right]$$

central difference; but we are writing all this at n+1th level and we could have written Crank-Nicolson, then we had to write

$$\left[\frac{T_{i+1}^{n+1} + T_{i+1}^n - 2T_i^{n+1} - 2T_i^n + T_{i-1}^{n+1} + T_{i-1}^n}{2} \right]$$

So, average of two time-level, we could have written, then also we would have gotten unknown terms on the right-hand side. But here we are writing directly at n+1th level so that you know from here also we can formulate the problem. Now, next step you can see

we have transferred Δt on right-hand side divided by $(\Delta x)^2$, $\frac{\alpha \Delta t}{(\Delta x)^2}$ this is F. So, we can write the equation.

We have brought all the n+1[th] level times on the left-hand side,

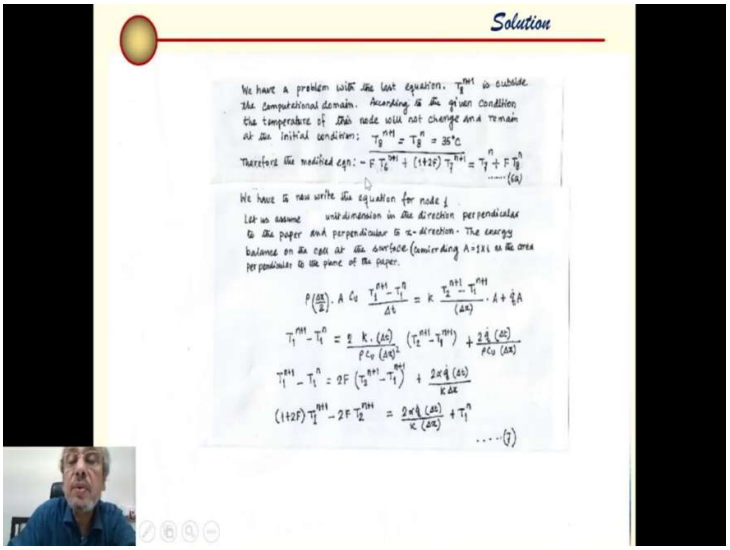
$$-FT_{i-1}^{n+1} + (1+2F)T_i^{n+1} - FT_{i+1}^{n+1} = T_i^n$$

and we can substitute here, i = 2, to i = 7. i = 2 means then it will be expressed in terms of T_1 and T_3, T_2 will be expressed; i = 3 means T_3 will be expressed in terms of T_2 and T_4; i = 4 means it will be expressed in terms of 3 and 5.

So, this way i = 5 means it will be expressed in terms of 4 and 6; i = 6 will be expressed in terms of 5 and 7; 7 will be expressed in terms of 6 and 8. So, we will get these six equations straight away, but we have to now implement the boundary condition. First boundary condition is very simple.

Because see we have written here T_8; but T_8 is not you know within the domain, it is falling outside the domain. But we will use that information on T_8 as the boundary condition, it has been told that after 25 say after this elapsed time; that means, after 25 seconds, you know this temperature will remain at the initial temperature. So, T_8 is known which will remain as the initial temperature.

(Refer Slide Time: 71:01)



So, we will write the last equation as this was

$$-FT_6^{n+1} + (1+2F)T_7^{n+1} - FT_8^{n+1} = T_7^n$$

This is now FT_8 is behaving as T_7 of the previous level. So, we will just substitute that. We are transferring this as T_8 at n[th] level. We are transferring to the known side.

So, left-hand side, unknown side we will be left with this term and this term. This is known term, now transferred to the right-hand side.

$$-FT_6^{n+1} + (1+2F)T_7^{n+1} = T_7^n + FT_8^n$$

Now, what we will do? We will apply this equation; that means, the thermal energy balance equation at this node. So, energy coming in, energy going out through conduction and the energy that is heating up this volume, the thermal energy contained in this volume. So, simply we have written that part. So, this is $\frac{\Delta x}{2}$ is the dimension in x-direction; y and z-direction dimension are unity. So, basically, we are saying A = 1x1

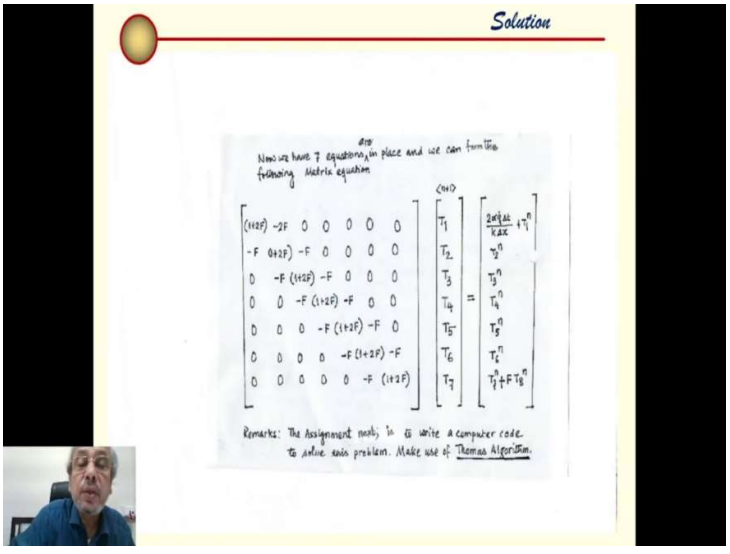
$$\rho\left(\frac{\Delta x}{2}\right)\cdot A\cdot C_v\cdot\frac{T_1^{n+1}-T_1^n}{\Delta t} = k\cdot\frac{T_2^{n+1}-T_1^{n+1}}{(\Delta x)}\cdot A + \dot{q}\cdot A$$

and here basically $\frac{T_2^{n+1}-T_1^{n+1}}{(\Delta x)}$ that will be the thermal gradient conduction gradient into again the cross-sectional area in z and y-direction. And $\dot{q}\cdot A$ is basically the thermal energy that is coming in. So, thermal energy that is coming in, the space energy is going up and thermal energy is being conducted. So, these three quantities are participating quantities and from there just next line following the algebra, we will be able to create

$$(1+2F)T_1^{n+1} - 2FT_2^{n+1} = \frac{2\alpha\dot{q}(\Delta t)}{k(\Delta x)} + T_1^n$$

So, at n+1[th] level T_1 and T_2, these are basically, you know this quantity on the right-hand side, this is α thermal diffusivity. This is $\dot{q}$ heat flux, Δt, Δx, k thermal conductivity, all are known and this is T_1 at the initial time level, this is also known. So, this can go to the right-hand side. First equation, then will be $(1+2F)T_1^{n+1} - 2FT_2^{n+1}$ equal to this entire term, which can be evaluated will be pushed to right-hand side.

(Refer Slide Time: 74:51)



We have exactly done that; $(1 + 2F), -2F$ and this it is coefficient of T_1 and this is coefficient of T_2. So, these two entries are there in coefficient matrix. T_1 and T_2, these are unknown terms and this known quantity from this previous equation.

This has come here. Now, from node number 2 to node number 7, all the equations we can write. Basically, this will read as T_1 into this quantity (-F) plus T_2 into this quantity (1+2F), plus T_3 into this quantity (-F) equal to T_2^n. So, all the equations now can be written in terms of coefficient matrix, unknown vector equal to known vector.

And as we discussed, the last equation should be again boundary condition that T_8 is getting involved, but T_8 is outside the domain and information is known that it is at initial temperature. So, that this is the last equation. Entire right-hand side all the quantities are known, we can evaluate them; T_1 to T_7 are unknown and this side F is a known parameter, (1+2F) is also when F is known, this is known quantity and this is (-F). So, this is a tridiagonal matrix.

In the coefficient matrix, all the terms are known. Right hand side resultant vector is also known, but this unknown vector is not known, T_1 to T_7. If we have just discussed Thomas algorithm, if we apply Thomas algorithm, we will be able to solve this.

Now, I have a request to the participating students that this problem is there. Please read it. I have formulated this problem on behalf of you. Try to understand and appreciate this. Then, write a code by yourself to solve this problem and find out the result T_1 to T_7 using Thomas algorithm. So, that's all for today.

Thank you very much. The class has been extended little beyond; but if you have paid good attention, you have learned a lot. And with confidence, you will be able to apply this in our subsequent exercises.

Thank you very much.