Computational Fluid Dynamics

Prof. Dr. Suman Chakraborty
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

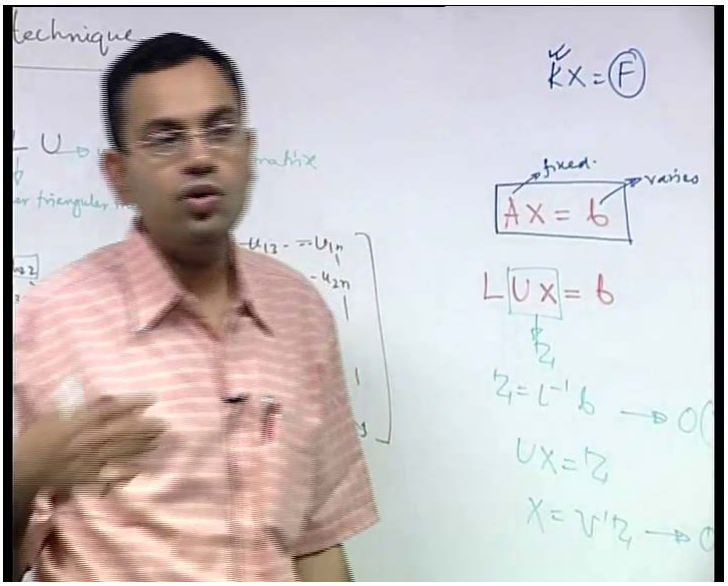
Lecture No. # 23
Solution of Systems of Linear Algebraic Equations:
Elimination Methods (contd.)

In the previous lecture, we were discussing about some aspects of the elimination methods and where we ended up, was the discussion on LU decomposition technique. To summarize what we discussed in the LU decomposition technique, we decomposed or factorize the co-efficient matrix a as a product of a lower triangular matrix and an upper triangular matrix.

The objective was to handle triangular matrices, so that you can come up with an algorithm, which is as efficient as the substitution method, the backward substitution method or an equivalent forward substitution method, which could be part of the Gaussian elimination method, but not the forward elimination part, which is of the order of n cube in terms of its computational cost.

Where we landed up is, that although, the forward substitution or backward substitution, either of these are of the order of n square, but the LU factorization itself has a cost of the order of n cubed, that makes the algorithm in terms of computational cost no better than the Gaussian elimination. But still in many cases, we prefer to use the LU decomposition as compared to the Gaussian elimination. What are those cases? So, you have to remember that in Gaussian elimination, there are two parts. One is forward elimination, another is backward substitution. Here, there are basically two parts. One is the LU factorization; other is the implementation of the forward substitution and backward substitution.

(Refer Slide Time: 02:18)



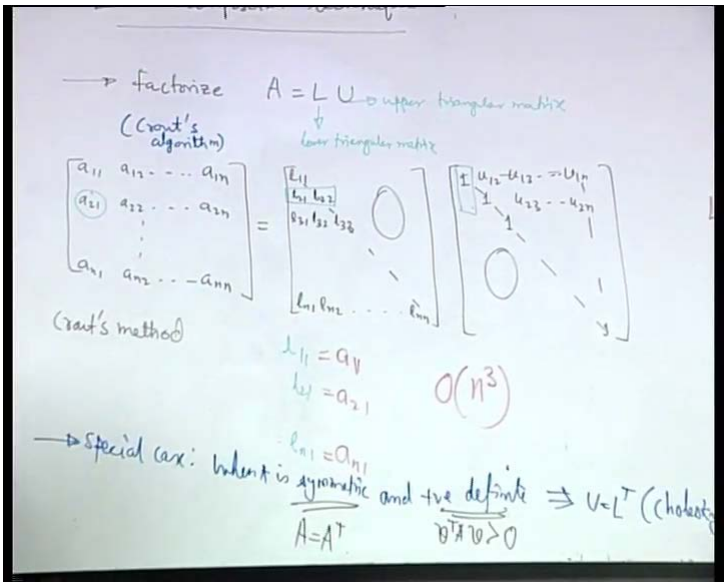
So, if you see, let us consider a case where you have a system AX equal to b. This type of system where A does not change, but b varies. Classical example may be a situation, where you have a particular layout of a spring mass system where you have different applied forces on which you try to solve the system or may be a structure, say you have a structure like this. Some structure where the structure remains the same. Only the load applied changes from one case to the other and you are testing the structure with different loads. Or you have an electrical circuit, where you change only the voltages, but the resistors making the circuit, they remain the same. So, if you just have an equivalent representation of, say i r equal to v where v changes, but r remains the same. Then, what is the current that changes because of the change in v, say if you want to find out that. So, with these examples, we are seeing that there are possibilities where there is a simple algebraic equation. The right hand side of which may change under different circumstances, but the co-efficient matrix does not change.

So, you have in general, k x equal to F of this form, where F is a forcing function, k is the stiffness matrix. So, to say just with an analogy of the finite element method, your load vector F changes for starting different cases, but k does not change. So, here k is equivalent to the co-efficient matrix A. So, when the co-efficient matrix A does not change, then what is the advantage that you are having? When you have to simulate different problem where the co-efficient matrix is the same, but the right hand side is different for different problems, then what is the advantage?

Advantage is that you have to factorize it only once because factorization into L and U does not depend on the right hand side. It does not depend on b, it depends on only what is A. So, if b varies in different cases, but A remains the same, then you have to pay the price of LU factorization only. Once you have done that, then for each and every individual problem, you can use that same factorization without having to do it again and again. Then, your cost is either forward substitution or backward substitution which is of the order of n square.

So, you can use LU decomposition technique under certain circumstances where or rather, it is preferable to use LU decomposition technique for those circumstances where you have changing right hand side, but the co-efficient matrix does not change. So, every time you do not have to pay the price of calculating L and U. For calculation of L and U, that is, for factorization, one can have a particular simple way of doing it which has a number of calculations a bit less than what is required in this formula. In this formulation, this is known as Crout's algorithm. So, to say what we discussed in the previous lecture about LU Factorization which is of the order of nQ, now that number of calculations may be reduced to some extent if A is a special type of a matrix. So, special case, when A is symmetric and positive definite.

(Refer Slide Time: 06:28)



In that case, you will have U as L transposes itself. This is known as Cholesky's LU factorization formula, but even then, although number of computations reduce that you

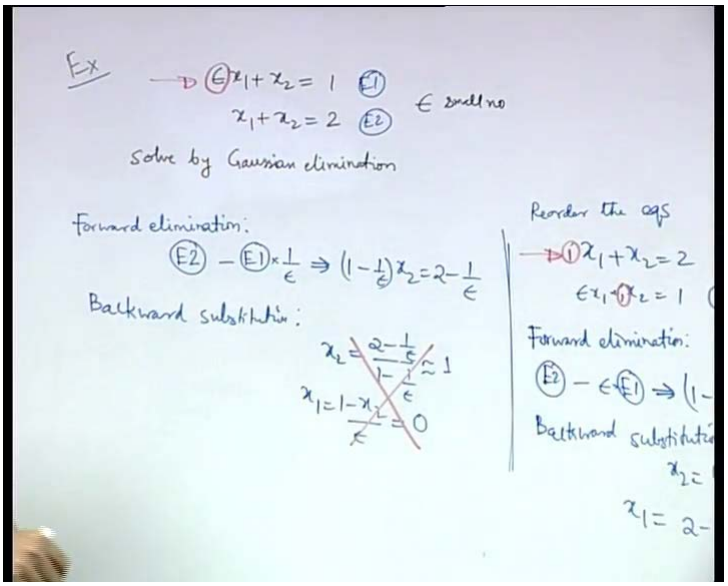
still have the computations for the factorization of the order of nQ. That does not change in terms of order. Although, the exact number of calculations roughly become half of the original numbers of calculations because here, you have to calculate L and U. Here, you have to calculate either of L or of U because one is the transpose of the other. So, numbers of calculations becomes half, but that does not change the order. It does not change from nQ to any other order.

Now, so, it is important to understand that, that can be applied when A is symmetric and positive definite, so symmetric. What do you mean by a symmetric matrix? When A and A transpose are the same and positive definite, if you identify a vector v, any vector v such that v transpose av greater than 0, then we call A as a positive definite matrix. So, v transposes a v, where v is a vector, any vector.

So, if A satisfies these two conditions, then only you can use this Cholesky's factorization principle. So, we have sent the Gaussian elimination. We have seen the Cholesky's rather LU decomposition method with the Crout's reduction or Crout's factorization formula and the Cholesky's factorization formula.

Now, before going ahead further, let us try to figure out, whether the Gaussian elimination method will work for all cases or not. We have seen that if it works how efficiently it works and that we have seen that what its computational complexity is. Is there any error involved or how the error is related to the implementation of the method, that we will come to know subsequently, but before doing that we have to understand that, will the method always work or not.

(Refer Slide Time: 10:25)



So, let us consider an example, say you have to solve a system of two equations solved by Gaussian elimination. Here, epsilon is a small number in the limit. It is 10ds to 0, but in the computer, you can take it as a number which is very small as compared to the precision that is available. So, let us try to go through this problem. Let us say, this is equation 1 and this is equation 2. So, the first step is the forward elimination.

If you want to eliminate x 1 from equation 2, then what would be the corresponding manipulation? It is equation 2 minus equation 1 into 1 by epsilon. So, that will mean 1 minus 1 by epsilon x 2 is equal to 2 minus 1 by epsilon. So, that is the consequence of forward elimination and backward substitution x 2 is 2 minus 1 by epsilon by 1 minus 1 by epsilon. So, what will be the value of x 2? So, you have to keep in mind that epsilon is a very small number. So, 1 by epsilon is a very large number minus 1 by epsilon is a very large negative number in comparison to that. There is no difference between 2 and 1. So, it is a minus of very large negative number. That means, a very large negative number divided by a very large negative number. So, that is approximately 1.

Then, x 1 will be 1 minus x 2 by epsilon. So, that will be 0. You can see that clearly this solution does not satisfy the equations. So, if you substitute x 1 equal to 0 and x 2 equal to 1, it satisfies equation 1, but not the equation 2. So, these are not the correct solution to test whether we get something different by reordering the equations. Let us reorder the

equations. So, x 1 plus x 2 equal to 2, that is, equation 1 and epsilon x 1 plus x 2 equal to 1 that is equation 2.

Then, let us go through the forward elimination step equation 2 minus epsilon into equation 1. So, backward substitution x 2 is 1 minus 2 epsilon by 1 minus epsilon. Epsilon is a small number. So, this is approximately 1 and x 2 is 2 minus epsilon. Sorry, 2 minus x 1 sorry x 1 is 2 minus x2, that is, 1.

So, this satisfies the given system of equations. So, the objective of this exercise is not just to see what the solution of this very simple system is, but to learn something out of it. So, what we saw here is that here, the pivotal co-efficient or the diagonal co-efficient is small and that created the problem. So, what is the origin of the problem? The original of the problem is a small value of the pivotal co-efficient or a small value of the diagonal entry.

Here, because of exchanging the equations, now the order of the equations is changed. So, equation 1 becomes this 1. Its diagonal is 1. Equation 2, it has its diagonal 1. So, none of the diagonals are very small numbers. So, what is the problem if the diagonal is a small number? So, we want the diagonal to be big enough in terms of magnitude. Later on, we will see that this leads to a condition, which is more formally known as a diagonally dominant system, that is, the diagonal element in terms of its magnitudes dominates over other sub-diagonal, off diagonal elements, but before going formal into that, we can informally get a feel of that one. So, why do we require the dominance of the diagonal or why do we require the diagonal to be large enough?

You can see that essentially, you have to divide by the diagonal during the forward elimination process. So, that if you remember, for example, in the Gaussian elimination, you have a division by a k. So, here that a k equal to 1 means, a 11, that is epsilon. So, you can see even in this formula where for manual implementation also, we are using the same scheme.

Now, if epsilon is a small number, then 1 by epsilon becomes a large number. So, division by a small number makes it so large that it can blow off and over weigh the effect of any other number. So, when it is 1 by epsilon, its effect has become so large that it can over weigh the effect of the difference in the other coefficients. So, whether the other co-efficient are 1 2 3 5 10 20 50, it does not make any difference because 1 by the

small number has become large enough to over weigh the effect of the differences in various coefficients. If you are not able to resolve the differences in various coefficients, then you are bound to get errors.

So, you have to have the method where the differences in the values of various co-efficient or the right hand sides, they need to be preserved that difference should not be nullified. Now, by here, by this small number division, the difference between 2 and 1 has been nullified. So, as if it does not matter whether the right hand side is 1 2 5 10 50 100 whatever, its 1 by epsilon that dominates.

So, you do not allow a small number to be there in the diagonal co-efficient because 1 by that will become a large number and that will over weigh the effect of any other co-efficient. That is why, if such is the case, you will always have a chance that the Gaussian elimination will give you erroneous solutions and the way out is, to reorder the equations before implementing the Gaussian elimination and that is known as Pivotization.

So, pivotization means, you reorder the equations to reassign a pivotal row. Here, the pivotal row was the first row. Now, if you interchange those two, then the previous second row becomes the pivotal row. The pivotal row is considered to be such that the corresponding diagonal element is the maximum in magnitude of all the possible options that you have.

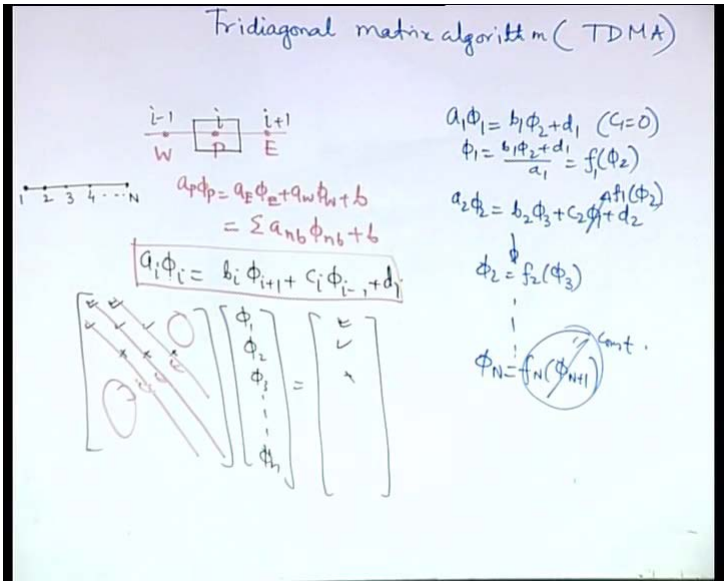
So, if you had for example, 10 equations. Out of that, consider that row as the pivotal row for which the first diagonal co-efficient is the maximum in terms of magnitude. Always remember, that it is a magnitude that we are looking for. Not maximum in terms of the sign number. It is the unsign, the magnitude because when you divide by a small number, it is the magnitude of the number that will become important, not whether it is plus or minus.

So, pivotization essentially is re-assigning the pivotal row in a way, that you have the corresponding pivotal co-efficient which is the diagonal co-efficient corresponding to that row, as the maximum possible in magnitude out of all options of the equations that you have. That will make sure, that you are having the best possible option out of your given choice for implementing the Gaussian elimination. Even then, it may be possible that it is not good enough. That even the best one has a very small diagonal co-efficient.

Then, such a system is prone to serious errors when implemented or solved by using the Gaussian elimination technique. That is one of the important things that we have to keep in mind. So, Gaussian elimination will not always work. There are issues of diagonal dominance and if those issues are satisfied, then only we expect that the Gaussian elimination will work.

So far, we have seen algorithms, where you have of the order of n cube or n square as the number of computations or the computational cost, but can we bring it down two of the order of n. That is one of the very important and interesting questions that one would like to answer because if that is possible that can reduce the number of computations to a significant extent and that is indeed possible. Not for all types of matrices, but for matrices which are tri-diagonal in nature.

(Refer Slide Time: 23:52)



The corresponding algorithm is known as tri-diagonal matrix algorithm or TDMA, is also known as Thomas algorithm. Let us first try to understand what is the motivation in cfd? This is a very important algorithm. Why it is because in cfd, it is not very uncommon to get matrix as a co-efficient matrix. Why? Let us consider a one-dimensional problem.

Let us say, that you have grid point P. You have its neighboring grid points, E and W. So, you have a P phi P is equal to a E phi E plus a W phi W plus b. In general, it is a summation of the co-efficient from its neighboring grid points. Of course, the source

term, you can write it in terms of indices. If you can assign indices, you can write if you call P as i E as i plus 1 and W as i minus 1. You can write a i phi i is equal to b i phi i plus 1 plus c i phi i minus 1 plus d i of this form where i is P i plus 1 is E i minus 1 is W.

This is logically obvious because in our discretization, we make sure that events at a grid point are influenced only by its immediate neighbors. Other neighbors do not come into the picture. If you use higher order interpolation techniques, then only other neighbors will come into the picture, but because of the linear profile assumption, only the immediate neighbors will come into the picture. So, you will have for a one-dimensional problem only two neighbors plus the grid point itself make 3 grid points involved at each time.

So, if you write it in a matrix form, something into phi 1 is something into phi 2. There is no phi 0. Let us say your grid point starts with 1, then 2, then 3, then 4 and in this way, upto n. So, this is your domain. So, phi 1 has phi 2. There is no phi 0. Nothing called phi 0. So, phi 1 and phi 2 are the 2 non 0 co-efficient for the first equation and there is something in the right hand side.

For the second equation, something into phi 2 is equal to something into phi 3 plus something into phi 1 plus something. So, the second equation involves non 0 co-efficient as phi 1, phi 2 and phi 3. So, phi 1, phi 2, phi 3 and something in the right hand side. Third equation will involve phi 2, phi 3 and phi 4. In this way, you will see that if you have considered all the equations, what are the non 0 terms which are involved? One is this diagonal term. This corresponding to phi 1, this corresponding to phi 2 and this corresponding to phi 3 like that. Then, the two immediate of diagonal terms and all others are 0. Such a matrix, such a co-efficient matrix is called as a tri-diagonal matrix where you have the main diagonal and the two immediate of diagonals.

So, any algorithm corresponding to the solution of such a system is known as the tri-diagonal matrix algorithm. So, the first thing that we have understood is why the importance of tri-diagonal matrix is there? It is for a discretized system of equations in one-dimensional problem.

Next is how it is possible that you could come down to computational complexities of the order of n? It is possible while considering that any matrix for its storage requirement, it requires two indices, i and j. Here, what we will do is, instead of using

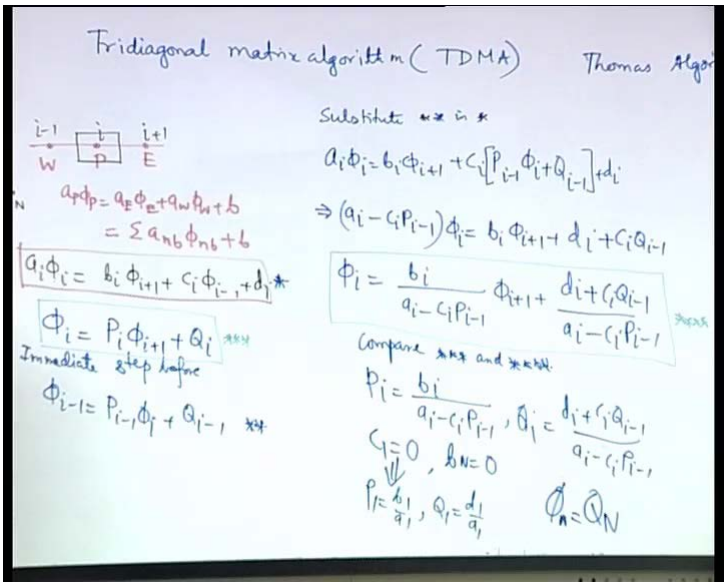
two indices i and j, we will just use one index for storing the entries corresponding to all these three diagonals together. That is how we will have a linear storage system, instead of having a two-dimensional array for storing the system, that is, the basic origin behind improving the computational problem of this algorithm.

So, let us try to figure out, that how we can do that. So, let us take a prototype equations of this particular form as a i phi i is equal to b i phi i plus 1 plus c i phi i minus 1 plus d i. Say, this is a equation. You have obtained out of one-dimensional discretization problem. Now, you can start from the left hand side to the right hand side.

So, your first equation will be, a 1 phi 1 is equal to b 1 phi 2 plus d 1 c 1 will be 0 because there is no phi 0. So, you can write phi 1 is equal to b 1 phi 2 plus d 1 by a 1. That means phi 1 is a function of linear function of phi 2. Second equation a 2 phi 2 is equal to b 2 phi 3 plus c 2 phi 1 plus d 2. In place of phi 1, you can substitute as function of phi 2. So, this will give you phi 2 as a function of phi 3. Linear function again in this way, you can use the previous step to write phi 3 as a function of phi 4, phi 4 as a function of phi 5. In this way, phi n will be a function of phi N plus 1, but there is nothing called as phi N plus 1. It is just a constant because there is nothing called N plus 1. N is the last grid point.

So, now, what we have to do is, we have to systematize this approach. So, as you know phi 1, you can know phi 2. From that if you know phi 2, you can know phi 3 from that. It is not known explicitly, but implicitly. If phi was known, then phi 2 would have been known. If phi 2 would have been known, phi 3 would have been known. In this way, if phi n would have been known, then phi N plus 1 would have been known, but there is nothing called phi N plus by 1. So, that means, in this way when you go to the last step, phi n is known.

(Refer Slide Time: 33:22)



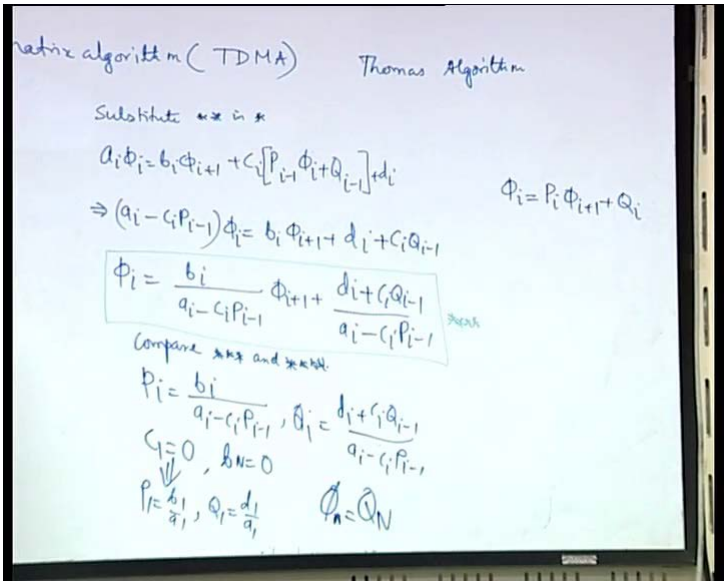
So, what we will do? We will consider that you have intermediate steps of this form phi i as P i phi i minus 1 plus Q i. This is the form that we look for. See phi 2 as a function of phi 3, sorry phi i plus 1, so phi 2 as a function of phi 3, phi 3 as a function of phi 4 and linear functions. So, linear function means of the form y equal to m x plus c of that form. So, y is a linear function of x, phi i is a linear function of phi i plus 1. So, this is the form of the linear function where i depends linearly on i plus 1, where this P i and Q i are not yet known to us.

What is the immediate preceding step? Immediate step before phi i minus 1 is equal to P i minus 1 phi i plus Q i minus 1. That is replaced i with i minus 1. Now, let us substitute this particular recurrence formula in the discretized equation. So, substitute in star, say this is double star substitute double star in star. So, a i phi i is equal to b i plus phi i plus 1 plus C i. In place of phi i minus 1, we will write P i minus 1 phi i plus Q i minus 1. So, form here, we get a i minus C i P i minus 1 phi i is equal to b i phi i plus 1 plus d i. So, phi i is equal to b i by a i C i a i minus C i P i minus 1 into phi i plus 1 plus d i C i into Q i. There is 1. C i into Q i plus 1 minus d i.

So, d i plus C i Q i minus 1 by a i minus C i P i minus 1. Now, you can see that this particular equation and the equation phi is equal to P i phi plus 1 plus Q i. These are of the same form. So, let us call it triple star. Let us give it 4 stars. So, compare these triple stars and 4 stars. So, what you get?

P i is equal to b i by a i minus C i P i minus 1 and Q i is equal to d i plus C i Q i minus 1 by a i minus C i P i minus 1, where you have to keep in mind that C 1 is equal to 0. Why because there is no phi 0 and b n is equal to 0 because there is no phi N plus 1. C 1 equal to 0 implies P 1 is b 1 by a 1 and Q 1 is d 1 by a 1. What is phi n? Phi n is P n phi N plus 1 plus Q N. There is no phi N plus 1. So, phi n is Q N because what is the formula that you are having phi i is equal to P i phi i plus 1 plus Q i.

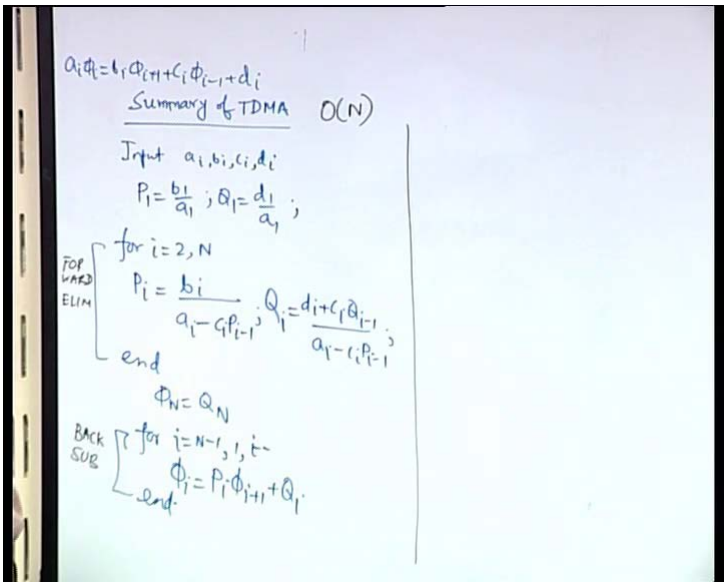
(Refer Slide Time: 40:19)



So, if you know phi N, then you can calculate phi N minus 1. If you know phi N minus 1, you can calculate phi N minus 2. So, that is your backward substitution. In this way, the first value of phi you calculate is phi N, then phi N minus 1, then phi N minus 2, phi N minus 3. In this way, you end up with phi 1. So, that is how you solve for the unknowns from phi 1 to phi N.

So, here also, it is logically like divided into two parts. Forward elimination, but forward elimination is this part and the remaining is backward substitution, where it is a straightforward use of this formula. So, based on this, let us now summarize the TDMA. So, first is you input a i b i c i d i that comes from your discretized equation. Then, you have P 1 is equal to b 1 by a 1, Q 1 is d 1 by a 1. Remember, the equation that you are looking for is, a i phi is equal to b i phi i plus 1 plus c i phi i minus 1 plus d i.

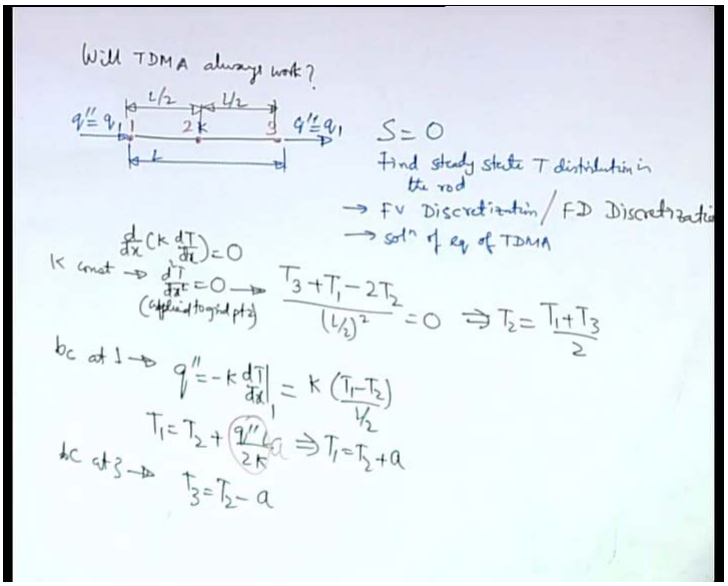
(Refer Slide Time: 41:18)



Then, for i equal to 2 to N, you can calculate P i. This is the recurrence relationship. So, if you know P 1, you can calculate P 2 and Q 2. If you know P 1 and Q 1, you can calculate P 3 and Q 3 and so on. Then, phi N is equal to Q N. Then, you have a, for loop in which you put the formula for phi i is equal to P i plus phi i plus 1 plus Q i. So, for i equal to N minus 1, it will start with phi N minus 1. Phi N is already known. It will end zp to 1 with i minus, that is you have an increment of minus 1. That is each i will be 1 less than the previous i. So, this is the forward elimination part. This is the backward substitution part.

You can see that the forward elimination part is having a complexity of the order of n because there is only 1 for loop there which runs upto the order of N. So, the TDMA has an algorithm complexity of the order of N, not N square or N cubed. That is a big advantage. Now, as for any other method, you would also like to know whether the TDMA will work for all cases or not. To do that, we will consider or to asses that, we will consider a very simple problem.

(Refer Slide Time: 45:06)



So, let us say that we have a rod one-dimensional system, where you have a heat flux at the left wall as given. Say, Q 1 heat flux at the right boundary is also given. Same as Q 1, the length of this one is L. Thermal conductivity is given. We will assume easy numerical value, so that we can evaluate that easily. So, the objective is to find out temperature distribution, steady state temperature distribution in the rod.

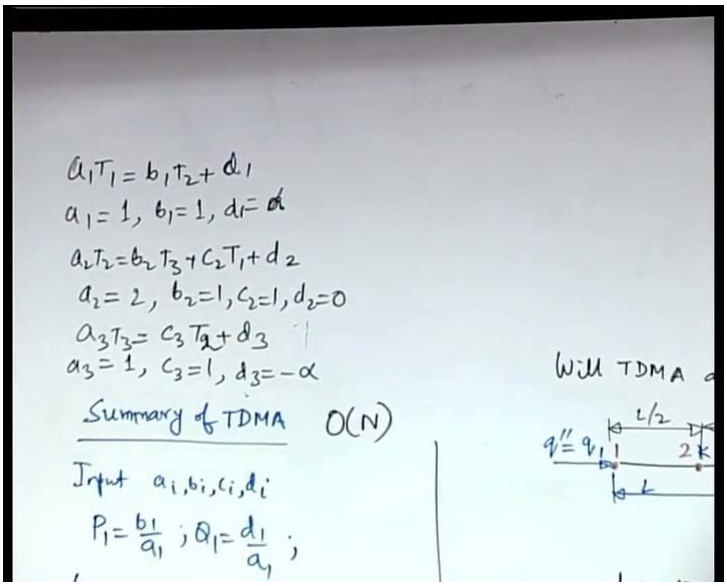
The steps will be finite volume discretization and solution of equation by TDMA. Let us consider that you have 3 grid points. You can also consider it to be equivalent finite discretization. If you want finite volume or finite difference, whatever you want, our focus here is not the discretization, but the solution. Let us take the second grid point at half way. So, this is L by 2 and this is L by 2. Let us form the discretization equation corresponding to the grids point 2. So, what is the governing equation d d x of k dT d x equal to 0.

If k is a constant, it is as good as d 2 T d x 2 equal to 0. So, if you consider a finite difference discretization, it is T 3 plus T 1 minus 2 T 2 by L by 2 square. That is equal to 0. Here, delta x is L by 2. This is applied to grid point 2. That means T 2 is T 1 plus T 3 by 2. For the boundary condition at 1 q is equal to minus k d T d x at 1, so k into T 1 minus T 2 by L by 2.

So, T 1 is equal to T 2 plus q L by 2 k. Let us consider this q lo by 2 k as number a for simplicity. So, T 1 is equal to T 2 plus a, then boundary condition at 3. In a very similar

way, T 3 is equal to T 2 minus a. It is physically obvious because heat is being transferred from higher temperature to lower temperature. So, T 3 will be less than T 2.

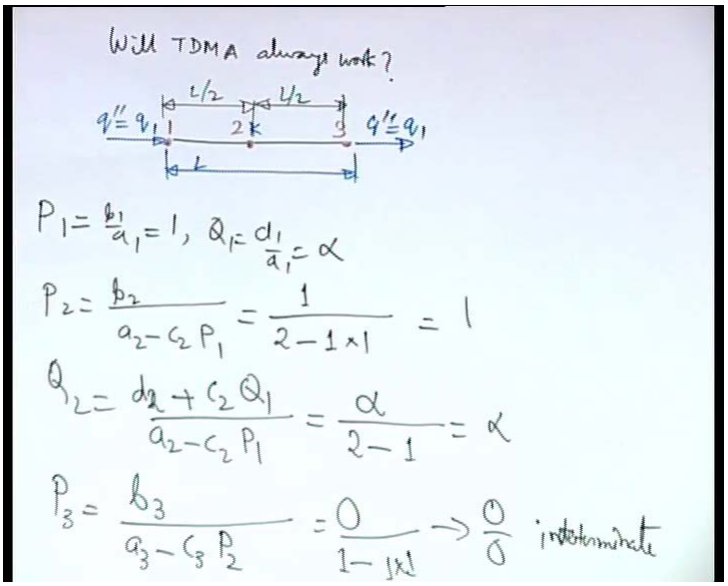
(Refer Slide Time: 51:20)



So, now, let us try to arrange these equations in this particular form a i phi i. So, a 1 T 1 is equal to b 1 T 2 plus d 1. What is a 1? a 1 is 1, b 1 is 1 and d 1 is in place of a. Let us give it a different name because a is already there. Say alpha, say, this is equal to alpha d 1 is alpha. Then, what is a 2? a 2 T 2 is equal to b 2 T 3 plus c 2 T 1 plus d 2 . So, what is a 2? a 2 is, you can write 2 T 2 and these as 1 and 1 in whatever way.

So, this equation, you can write as 2 T 2 equal to T 1 plus T 3 also. So, a 2 equal to 2 b 2 equal to 1 c 2 equal to 1 d 2 equal to 0. Then, for the third point, a 3 T 3 is equal to c 3 T 1 plus d 3. What is a 3 1 c 3 1 d 3 minus alpha. So, these are our 3 equations, c 3 c 3 d, that is, T 2.

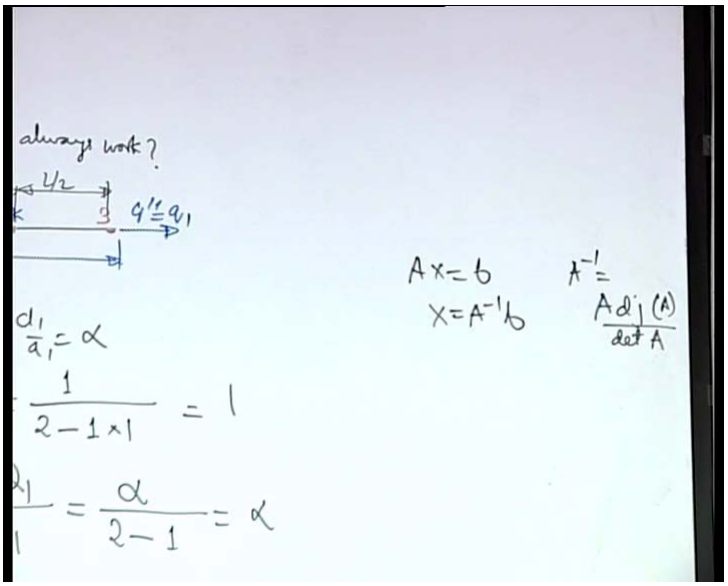
(Refer Slide Time: 53:46)



Now, you can go through that TDMA as it is there. Here, P 1 is b 1 by a 1. So, P 1 is b 1 by a 1, that is, equal to 1. Q 1 is d 1 by a 1 that is equal to alpha. Then, P 2 is b 2 by a 2 minus c 2 P 1. So, b 2 is 1 by a 2, 2 minus c 2 P 1, 1 Q 2 is equal to d 1. Sorry, d 2 plus c 2 Q 1 by a 2 minus c 2 P 1. So, what is d 2? d 2 is 0, c 2 is 1, Q 1 is alpha, so alpha by a 2 2 minus c 2 1 into p 1 1. Then, P 3 b 3 by a 3 minus c 3 P 2 b 3 is 0 divided by, let us see what it is. There a 3 1 minus c 3, c 3 is 1 into P 2 1. So, it is 0 by 0 from indeterminate.

So, you can see that the calculation of TDMA breaks down here. So, TDMA does not work for this case. Why it does not work for this case? If you consider the co-efficient matrix, you will find what the characteristic of the co-efficient matrix is. Here, determinant of the co-efficient matrix is 0 that you can easily verify. So, it is a singular co-efficient matrix.

(Refer Slide Time: 56:26)



So, remember that in whatever may the method that we are using for solution in A X equal to b, X equal to A inverse b. Inverse of a matrix is just like a sort of division. It is a systematic division considering a collection of number instead of a single number. So, if the inverse of A is what Adj joint A by determinant A.

So, if the determinant A is 0, if it is singular, then you can see a inverse as a problem. Physically, why that the TDMA does not work here. I am giving this problem as an example because we have referred to this example earlier. You have this as actually an imposed boundary value problem because you are having 2 flux boundary conditions, where the fluxes are equal. Of course, they should be equal because of one-dimensional steady state heat conduction requirement, but that is already inbuilt with the governing equation. It is no new information.

So, although you have specified fluxes at the 2 boundaries and they are equal. So, it is physically consistent, but it is not giving you any additional information corresponding to the boundary condition. So, that gives rise to the singularity of a system. So, you can see an ill posed boundary value problem if no where detected, otherwise will face trouble when you are coming to solve the system of corresponding discretized algebraic equations.

So, if the physics is somewhere erroneous in terms of describing the problem, somewhere mathematics will catch it. That is the beauty of mathematics that if you are

unable to represent the right physics through your mathematics, somewhere mathematics itself will interfere and detect itself, that there is something wrong with the physics.

So, TDMA may not work always. So, we will stop here today. In the next class, what we will do is, till now we have looked into the implementation of different elimination methods. In the next class, we will do an error analysis. We will try to figure out that what are the corresponding errors possible to be incurred and how we can quantify that in the elimination methods. Thank you.