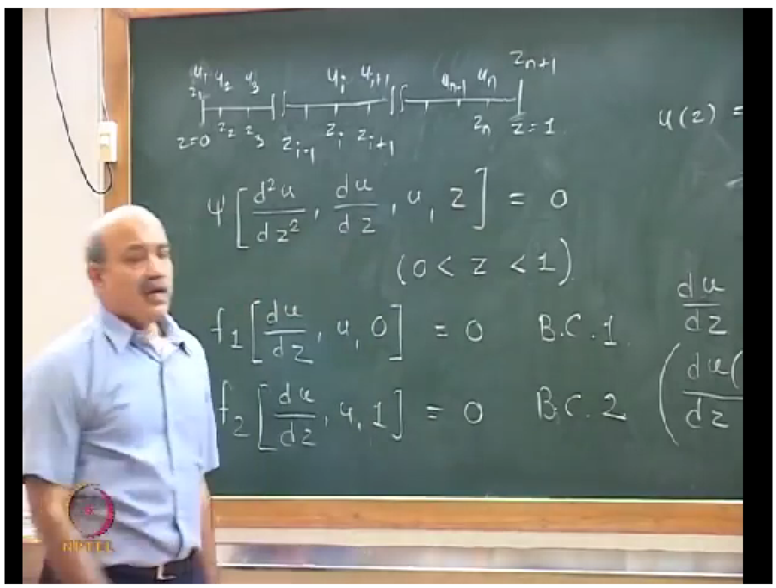
Advanced Numerical Analysis

Prof. Sachin Patwardhan
Department of Chemical Engineering
Indian Institute of Technology - Bombay

Lecture – 16
Orthogonal Collocations Method for Solving ODE - BVPs and PDEs

So we have been looking at problem of solving ordinary differential equation subject to boundary conditions using method of orthogonal collocations. So this is based on using interpolating polynomial and this interpolating polynomial is used over the domain of interest and this interpolating polynomial we have then used to discretize the problem, discretize the boundary value problem. So let us take a quit recap of what we have achieved till now.

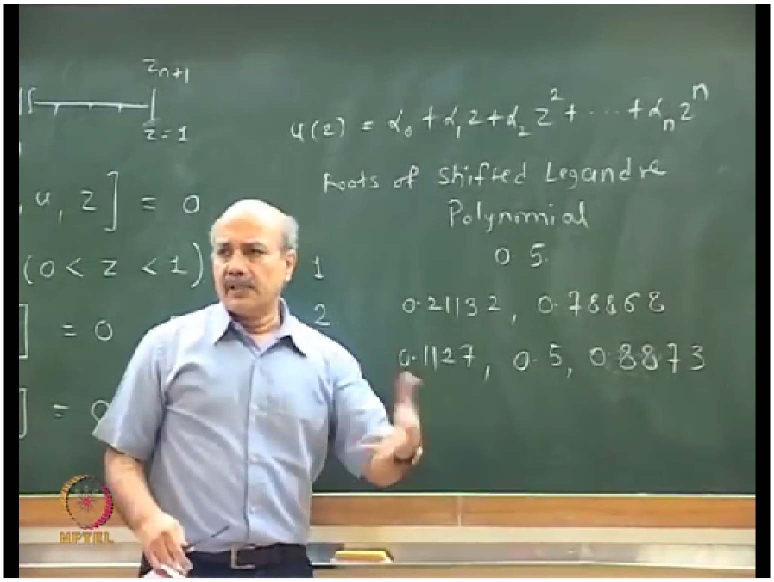
(Refer Slide Time: 01:01)



I have been looking at this problem of a general second order boundary value problem which is dz square, psi is a general function du/dz, u and z=0. Now this, this holds, this equation is supposed to hold over a domain 0 z<1. So here u is self dependent variable, it could be temperature, it could be concentration, whatever, whatever is the variable of interest. So we have this generic second order ordinary friction equation and then I have 2 boundary conditions, I have 2 boundaries.

So these are f1, so this is my boundary condition 1 and then my second boundary condition is du/dz. So I have this generic problem of second order, solving second order ordinary differential equation subject to these 2 boundary conditions, one at z=0, the other one at z=1, okay. So if I draw this on this domain, we have this domain here and we have already set the convention of (()) (02:55) so we have this domain here, this is z=0 to z=1, okay and then we have a solution, we have polynomial solution, interpolating polynomial solution which we have assumed.

(Refer Slide Time: 03:12)



So this is uz=alpha 0+alpha 1z+alpha 2z square+alpha nz to the power n, this is, this is the interpolating polynomial which is the proposed approximate solution and we have this convection of deciding or calling solution at certain points which are called as collocation points or the grid points. So in this context I want to call them as collocation points. So we have this collocation points which are numbered z1 z2 z3 in general this is zi, this is zi+1, this is zi-1 and the final one is, the final point is called as zn+1.

So we have this n+1 collocation points numbered from z1 z2 z3 up to zn+1, so these are collocation points, okay. Now these collocations points, they cannot be equi-spaced, okay. We have looked at finite difference method. In finite difference method, we looked at 2 options. One was the grid points as they were called in the finite difference method. The grid points could be equi-spaced; they could be non-equi-spaced.

In this case, though in principle, no one stops you from taking equi-spaced points. We are going to look at these collocation points chosen in a particular way. These collocations points are going to be chosen at the roots of shifted Legandre polynomials, okay. So these are going to be, these collocation points are going to be chosen at the roots of shifted Legendre polynomial given in the lecture notes. So I just list them here. So for example …okay.

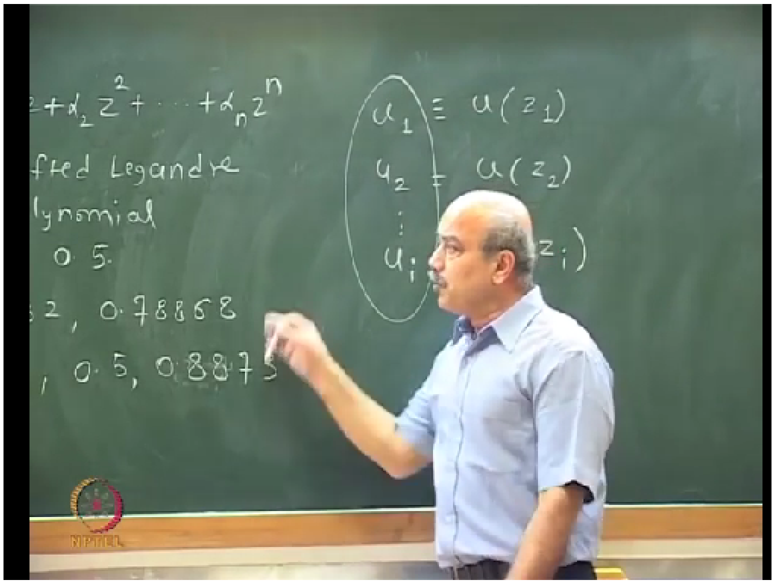
If I take the first order polynomial, then, so first order polynomial, then the root is at 0.5, okay. If I take the second order polynomial, then the root is at 0.21132 and 0.78868. If I take third order polynomial, then I have 3 roots 0.31127, 0.5, and 0.8873 and so on. So if you look at the standard text books, you will get these roots of the shifted Legandre polynomials and I am going to place these collocation points, I am going to place these collocation points, okay, at the roots of this shifted Legandre polynomial.

Which means if I happen to choose 3 collocation points in this domain, okay, the first one, the first one of course z1 will be 0, the second one will be placed at 0.1127, this is scaled, this domain is scaled between 0-1. Typically, if it is length, you can divide by that length scale to 0-1. So at point 1127 will be my second point. My third point will be at 0.5. My fourth point will be at in the 0.8873 and when the last point, the fixed point will be boundary z=1, okay.

So likewise here, in the accompanying notes, I have listed roots up to seventh order, okay and you will get, if you want to know about five order polynomials, you will get that in the literature. But typically it suffices use third, fourth, fifth or sixth order polynomials and if you want to have more collocation points, then typically what we do is, we do orthogonal collocation of finite elements which means we divide it into sub-elements and within that element, we define collocation points, okay.

So we will mainly go but if you want 50 collocation points, we do not do it by taking the 50th order polynomial, we take a 50th order polynomial, divide this domain into 10 segments and place the roots inside in each subdomain, okay, but that you will not be discussing now. We will be looking more at a single polynomial being chosen, okay. So let us do a quick recap of what we have done till now. We want to find the solution, approximate solution and then this approximate solution I have (()) (08:43).

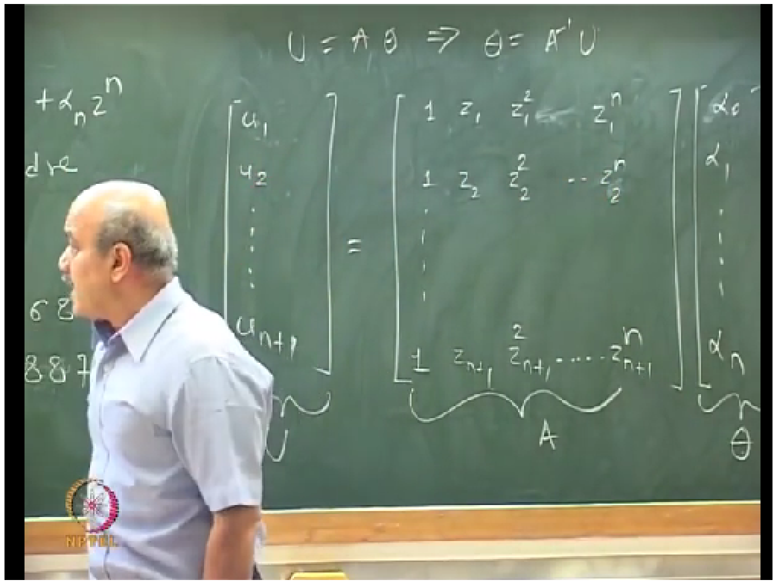
(Refer Slide Time: 08:46)



So my u1 corresponds to u, that is this approximate solution, computate at z1, okay and u2 is u computate at z2 at the second collocation point, okay and so on. So in general ui corresponds to u at z=zi, okay. Now what we said in the last lecture is that we would like to express these coefficients alpha 0, alpha 1, alpha 2, alpha n, okay, these are unknowns, okay. This is the proposed approximate solution for this ordinary differential equation, okay and this coefficients, I want to express in terms of u1, u2, u3, u4 and so on, okay.

So what is known to me here, what is known to me here is, I have chosen the collocation points. So the collocation points are known to me, okay. So these locations that is z1=0 to let us say if you take 3 points, z2=0.1127, these locations are known to me, okay. Now what we have done in the last lecture is like this, okay. To get unknowns transformed from alpha 0 to alpha n, okay. We wrote this equation at n+1 collocation points.

(Refer Slide Time: 10:30)

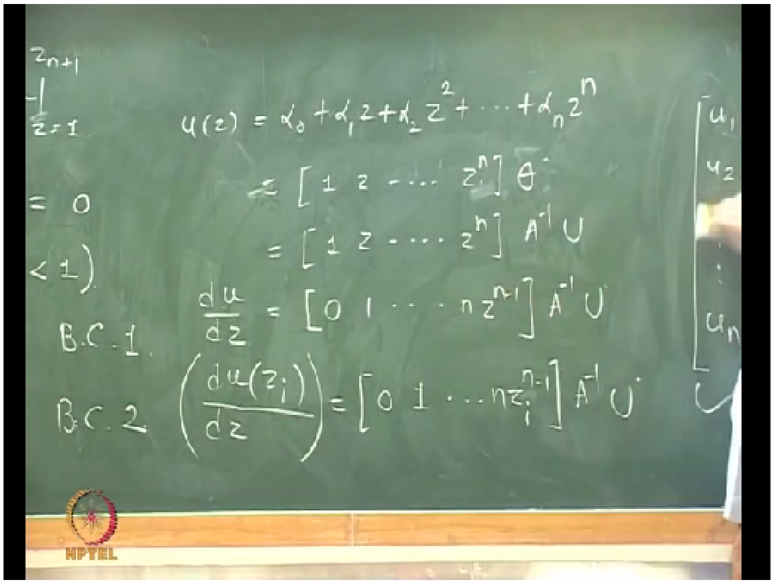


So I wrote this equation u1 u2 up to ui+1. So ui+1 is at the last point, okay and this will be 1 z1 z1 square up to z1 raise to n, 1 z2 z2 square up to z2 raise to n and so on, okay and we have this 1 zn+1; alpha 0, alpha 1 up to alpha n, okay. I rotate in this form. This is my, I define this matrix as A matrix if you recall, I called this as vector theta and these are my unknowns, these were represented as U, okay.

So to eliminate to express this alpha 1, alpha 2, alpha 3, alpha 4 in terms of unknowns u1, u2, u3, u4, well the problem here is that u1, u2, u3, u4 are actually solutions of this ordinary differential equation, okay at the collocation points. So this u1, u2, u3, u4 are not known to us, they will be known to us if you solve the differential equation, okay. So here is a double trouble, we do not know u1 to un+1. We do not know alpha 0 to alpha n.

But we want to transform the problems from bring it as unknowns to these unknowns, okay. So the way it has been done is to write this equation U=A theta, okay, this implies that theta=A inverse U, okay. Theta=A inverse U. Next what we have done is we need, we need derivatives of this function to be evaluated. See because here in this equation, you have d2u/dz square, you have du/dz, so I need these 2 equations to be evaluated, right. So for this what I have done is, so in the last class I derived this expression just to recall, I will not derive it again, I just write the expression, okay.

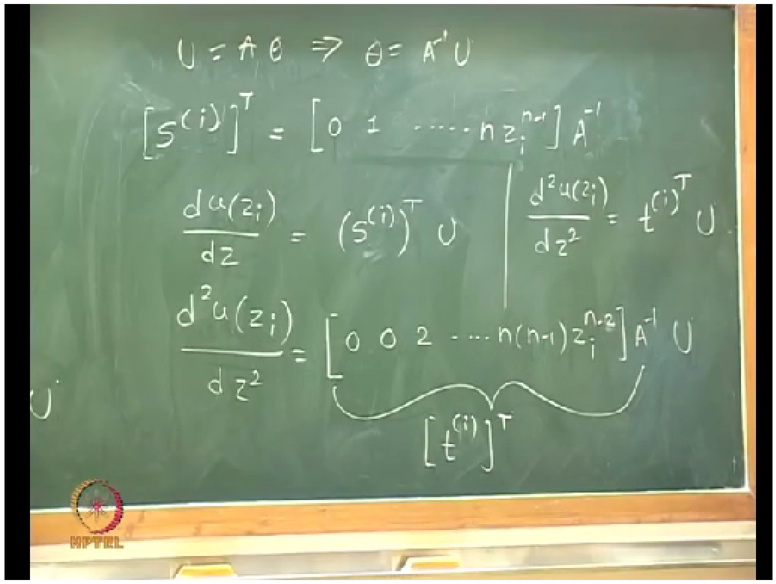
(Refer Slide Time: 13:24)



So we said that, we wrote this as 1 z z to the power n*theta, we wrote this as in the product of 2 vectors. One vector is 1-zn, okay and times theta. What is theta vector, this is theta vector, okay, this is theta vector and then we said that this is nothing but 1 z z to the power n*A inverse U. Theta will be replaced by A inverse U, okay. So in that instead of unknown as theta, we have now unknown as U, okay and then I want to write du/dz, okay. I wanted du/dz. If you take du/dz, this vector becomes 0 z up to nz to the power n-1 A inverse U, right.

This vector is just 0 to that and then I want to evaluate the derivative, I want to evaluate the derivative at these collocation points. So if I want to be further derivative at the collocation points, this becomes, so du/dz at z=zi, okay, this is equal to 0 1 nzi raise to n-1 A inverse U, okay. So this is the expression that we derived for the first derivative, okay. Similarly, we derive the expression for the second derivative, okay. We derive the expression for the second derivative. What is this expression, okay.

(Refer Slide Time: 15:46)



Before I move to the second derivative, this vector, okay, I call this vector as Si, I call this vector as Si transpose, okay. This Si transpose was defined as 0 1… This matrix, this A matrix is known to us, okay. This A matrix is known to us. So A inverse is known to us, A A matrix can be computed using because since we know z1 z2 z3, we can compute A matrix, we can compute A inverse, okay. So this, this matrix is known. Since we know zi, we also know this row, okay.

So this row times this matrix, this will be row vector, that row vector I am calling it as, okay, Si transpose. So my d … okay. My equation assumes the form du/dz=Si transpose U, okay. So likewise I also derive, I also derive expression for the second derivative d2u/dzi dz square, okay and that turned out to be I just give the final expression because we have derived last time, so it turned out to be 0 0 2, okay. We derived generic expression for the second derivative, okay.

Just, just look here, this is the first derivative. If I differentiate this with respect to z again, you will get 0 0, n-1 will come here, okay. This is just differentiation of this. This is a constant matrix. So this expression naturally follows from this and this is what we have got. We decided to call this particular vector as ti. I decided to call this particular vector as ti transpose*U, okay. So which means I have an expression that is d2u at zi dz square=ti transpose*U, okay.

I have expression. So what I have achieved, what I have done is I have expressed the derivative at a particular point at z=zi, okay as a vector*unknowns. What are the unknowns? Unknowns are values, the dependent variable takes at the collocation points, u1 u2 u3 u4, okay. So this is the derivative at this point is some linear combination of this vector u/2, okay. See what is the difference here. So when are u1 to u2 and u3 defined, so this is u1, this is u2, this is u3 and so on.

This is ui, ui+1. So this is zn, so this is un, this will be un-1 and so on, okay. So what is my u vector. My u vector consists of this dependent variable values at point 1 2 3 4 5 6 7 8, okay. So this entire vector is that u vector, okay. See what has happened. When you do finite difference, when you do finite difference, okay, you express the local derivative using only neighbouring points, or you express the second derivative only using neighbouring points.

Whereas here, the first derivative or the second derivative is a linear combination of entire, okay, u1 to un. So this is the difference. This is the main difference, okay. Everything comes include, everything comes into the picture, okay. So this is much much better way of finding the derivative then taking a local derivative, okay and then this term you are going to use to formulate the problem, okay.

So last time we stopped here. Let us now actually form the, substitute these values at the grid points and come up with the equation that it could be solved to get this equal to un+1. We still do not know what are the values of un, b1, un+1. All that we have achieved till now is to express these derivatives in terms of the unknown variables. What are the unknown variables u1 u2 u3 up to un+1 which are values, the dependent variable takes at the collocation points, okay.

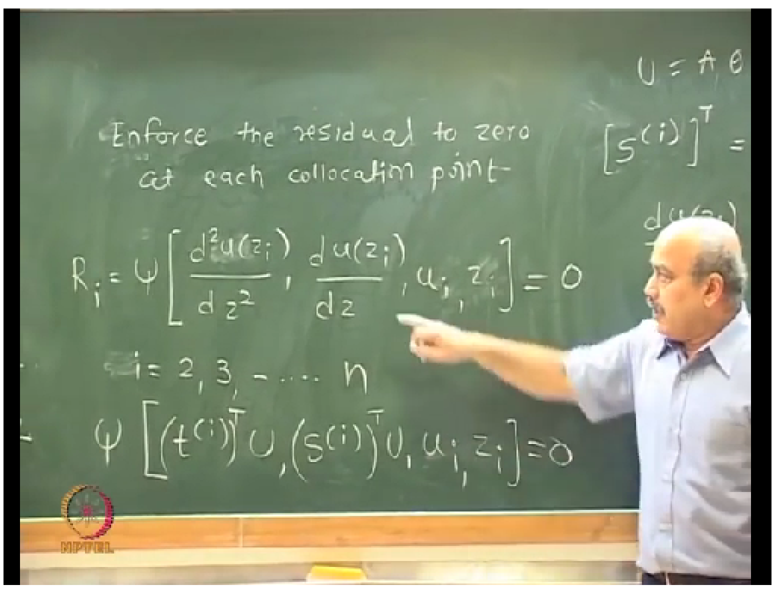
Now let us see how to solve the problem. Now I want to solve this problem, okay. So actually, see ordinary differential equation, where is it defined? It is defined on the domain, yes, it is defined on the domains 0-1. So it should hold, where should it hold? Everywhere. Ideally, the true solution, okay, the true solution u z, okay, this is not the true solution, this is an approximate solution.

The true solution should actually hold at every point in this domain, that is what it says, okay. Now when we solve it by orthogonal collocation, we are going to say that well the approximate solution should hold only at the collocation points, okay. Right now we are not saying anything what happens in between. At the collocation point, this equation should be satisfied. At the collocation point, this equation should be satisfied, okay. So this is discretization.

Actually we are looking at only finite number of points where the equation should hold. The original equation should hold everywhere, okay. So what is going to happen now is because of this approximation, this ordinary differential equation will get transformed into set of nonlinear or linear algebraic equations, okay. So this should hold inside the domain. At the boundary point, what should happen?

This equation should hold at first boundary, this equation should at the second boundary. So these equations, okay, enforcing this equal to 0 at each of the collocation points, will give you a set of equations plus these 2 will give you 2 more equations, we will have number of equations equal to number of unknowns and then we are going to solve them, okay. So now let us (()) (23:43) solving the problem.

(Refer Slide Time: 23:58)



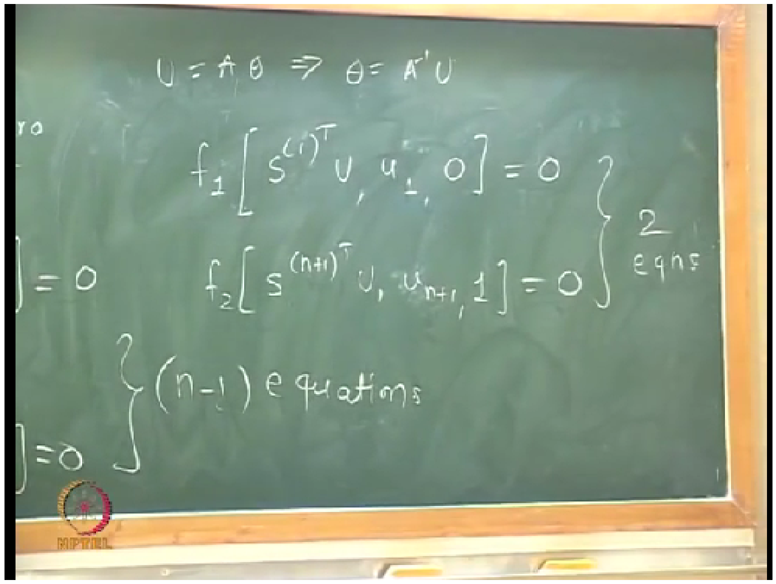
So the game plan is, enforce, so this has got residual, I just write down what, you will understand why am I calling it residual, to 0 at each collocation point. So that means what I am going to do? I am going to solve this equation. This is called residual psi, okay. Now d2uzi/dz square duzi/dz uzi, okay or in our case uzi is nothing but uizi should be equal to 0. This term I am going to call a residual. I want this equation to hold at every grid point, whichever grid points, for i=2, 3 up to n, okay. I want this to hold at…, okay.

Now if I substitute for expression that I have got, see the derivatives are now expressed in terms of algebraic expressions, okay. I want to replace it. So this actually means, I want to solve for ti transpose U Si transpose U, ui, zi=0, you get this equation, here, okay. See du/dz square, I have replaced by equivalent algebraic approximation, okay, sorry d2u/dz square, I have replaced by appropriate algebraic approximation.

Du/dz, I have replaced by appropriate algebraic approximation, okay. So this differential equation is now converted into an algebraic equation. How many such equations we have got now? We have got 2, i=2 3 4 up to n. So how many equations n-1. We have got n-1 starting from 2 to n, we have got n-1 equations, okay. We are going to set this residual equal to 0 at each of the collocation points, internal collocation points, okay. Now what about boundary conditions?

See how many unknowns are there, u1 u2 u3 up to un+1, okay. So how many equations you need? You need n+1 equations to solve it, okay. How many equations we have got till now? N-1. So we need 2 more equations. Those 2 equations are going to come from boundary conditions, okay. So the boundary conditions will give you additional 2 equations, that completes your set, we get n+1 equations, you get n+1 unknowns and then you are solving the property. We will solve them using…, okay. So let us write that to Gaussian equations, okay.

(Refer Slide Time: 27:35)



So the next equation is f1, okay, S1 transpose U, u1 at z=0=0, this is my first equation. This is my first equation and f2Sn+1 transpose U, un+1 at 1=0, okay. See now you have these 2 equations, okay and these n-1 equations, okay. These n-1 equations together with these 2 equations forms the set of n+1 equations. N+1 equations in n+1 unknowns, we need to solve them simultaneously, okay.

If these happen to be linear algebraic equations, we can solve them numerically. If they happen to be non-linear algebraic equations, you have to solve them reiteratively using some reiterative method, okay. So what we have done here? We have achieved transformation of the problem, of a boundary value problem, okay, from a differential equation to a set of algebraic equations, okay, using approximation theory. What method in approximation theory we will use, we will use interpolation, interpolating polynomial, okay.

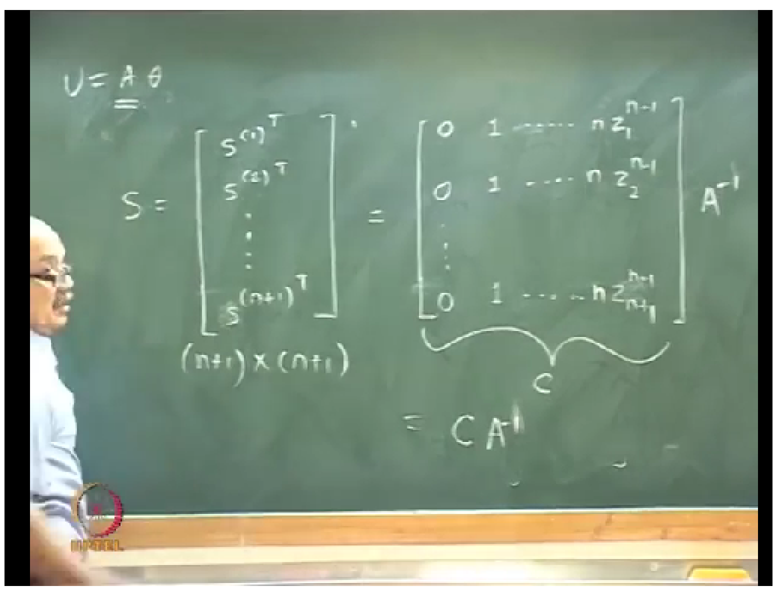
So these equations then you can solve it using the standard tools like Newtons method, Newton–Raphson or successive substitution, whatever. Whatever is suitable, that you can use to solve this particular problem afterwards, okay. Let us take a specific example that will be easier for you to understand. Before that, I just for the sake of convenience, I want to define 2 matrices, using this S vectors and using this t vectors, I am going to define 2 matrices S and t.

I am going to give you a method to compute them very easily, okay. So we will define these 2 matrices and then for a given number of collocation points, one has to first concept these matrices and then use them to formulate your equations. One thing which I would like to bring to your notice here is that in every equation, these are dense equations, these dense equations. In every equation, okay, u1 to un+1 will appear, okay.

Because the derivatives are approximating not locally but using all the points in their domain, okay. Since the derivatives are approximated using all the points, these equations will be dense, okay. So this is something different from, if you finite difference, okay, only 2 neighbouring variables will appear in 1 particular equations. Here it is not like that. Every variable will appear in every equation.

Particularly if you take a single polynomial over the entire row, okay. So that is the big difference, okay. So let me define this matrix S and T and then we will take this particular example that we have been using quite often or we will be using one example in the course which is Tubular Reactor with Axial Mixing. So before we do that, let me define these matrices.

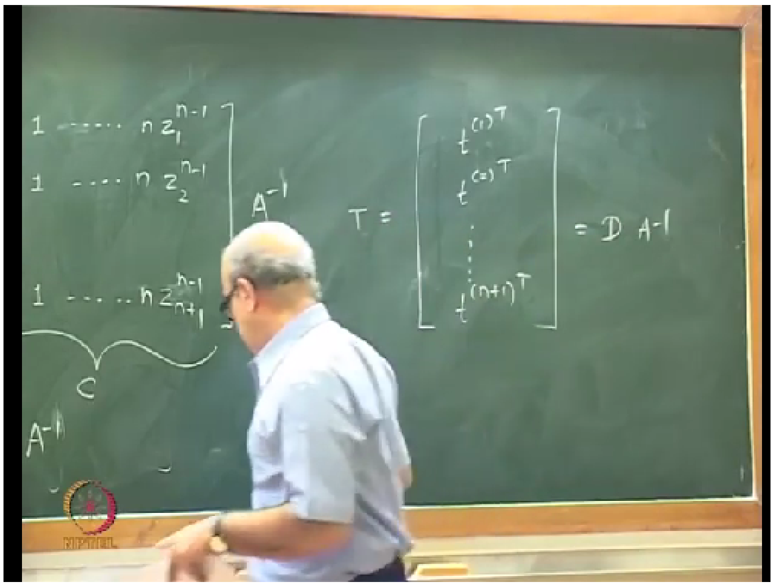
(Refer Slide Time: 31:31)



So this S matrix is going to be defined like this. It consists of S1 transpose, S2 transpose. So here S superscript 1 implies it is an actual vector, first vector, second vector, third vector and so on. So this is going to be Sn+1 transpose, okay. So this is the n+1*n+1 vector, okay. This is the n+1*n+1 vector, okay. It is very easy to show that this can be computed by looking at this another matrix 0 1 ...

So if I decide to call this matrix as say C matrix, then this is equal to C*A inverse. Well how was our A matrix defined? A matrix was defined, this keep it in background that you have this equation that U=A theta, okay, U=A theta. So if A matrix is defined … and so on. So likewise I am also going to define this T matrix.

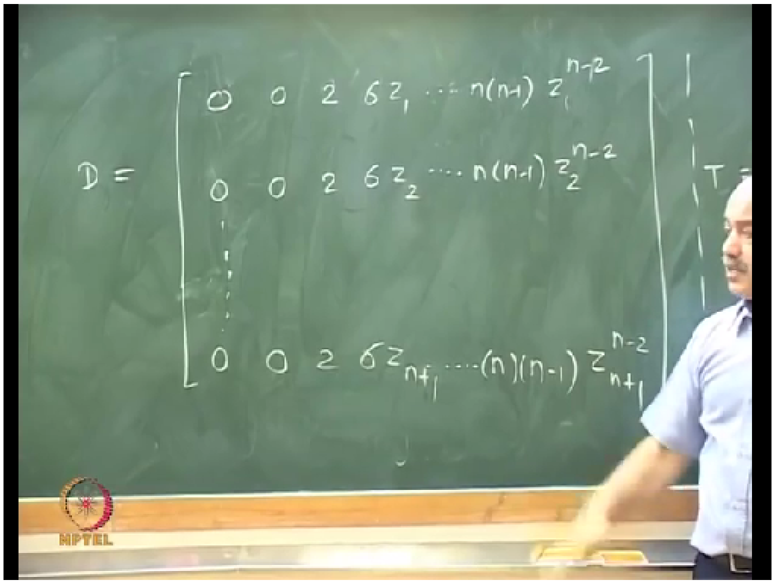
(Refer Slide Time: 33:47)



I am going to define this T matrix. This T matrix will consist of …, okay. So I have stacked up this row vectors, I have stacked up this row vectors, okay, to create this matrix, okay. Why this polynotation? Because in our course, whenever we are defining a vector, it is a column vector. When I want to make it row vector, I am making a column vector and putting it as a transpose, that is why this notation which we are getting, okay and then you can show that the best matrix is equal to, this can be very easily computed using D*A inverse and what is this D matrix, okay.

"Professor - student conversation starts" This is n+1*n+1. This is also n+1, yes. This is also n+1*… This is also n+1*n+1 times, yes. **"Professor - student conversation ends"** So let us write this D matrix, D matrix is slight modification of this matrix.

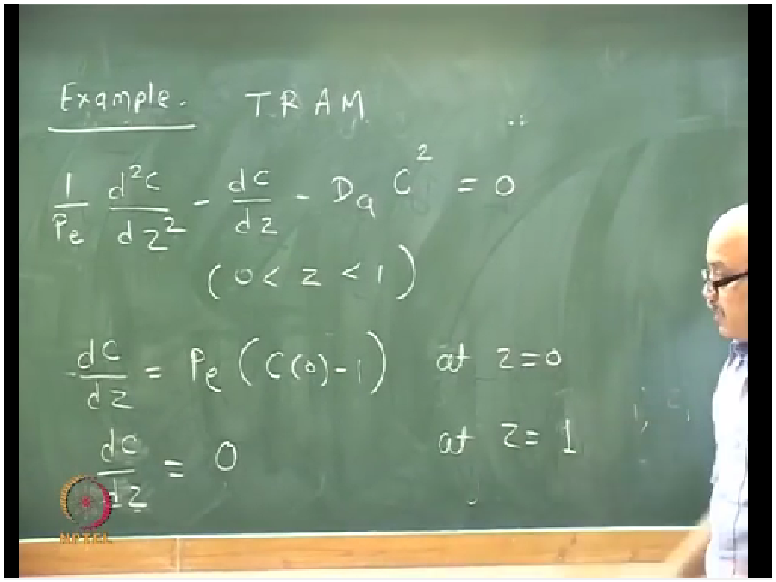
(Refer Slide Time: 35:13)



So D matrix will look like this. D matrix will be 0 0 2 6 z1… This is n-2, this is n-2 and so on. So this is nothing but… okay. This D matrix will be stacked vectors, okay. These are nothing but Ti vectors because they will add different collocation points and this together A inverse was multiplied by A inverse is going to give me D matrix, okay. So I need to create, there is a preparation for solving this problem, okay. I need to create S and T matrices, okay by choosing collocation points. Once I choose collocation points, okay, 3 4 5 6, whatever.

Once I choose the collocation points, I can first find out A matrix. Once I find out A matrix, I can find out A inverse and then I can define C and D matrices and from that I can get S and T matrices. Once I get S and T matrices, I am going to use rows of this matrices to discretize my ordinary differential equation and convert it into algebraic equations. Let us take a specific problem, then you will understand it better, okay.

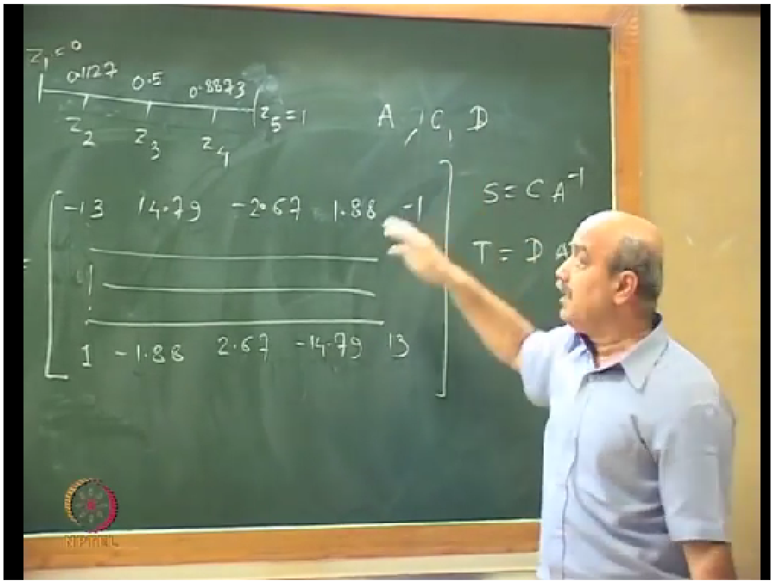
(Refer Slide Time: 38:04)



Okay, so this is our Tubular Reactor with Axial Mixing, okay. This is the problem which we have looked at earlier when we studied finite difference method, okay. The associated ordinary differential equation is 1/Peclet number*d2C/dz square-dC/dz-DaC square=0. So this is the ordinary differential equation that should hold between 0 z 1. Then you have 2 conditions, okay. You have 2 conditions, that is dC/dz, okay=Pe*C0-1 and this should happen at z=0 and dC/dz=0 at, dC/dz at z=0=0, okay.

This is the second condition that we have. 2 boundary conditions and…**"Professor - student conversation starts**" Second boundary (()) (39:56) yes, second boundary condition should hold at z=1. The second boundary condition should hold at z=1. So this is my problem, this I want to discretize, okay. **"Professor - student conversation ends"** Let us say I have chosen 3 internal points, okay.

(Refer Slide Time: 40:17)



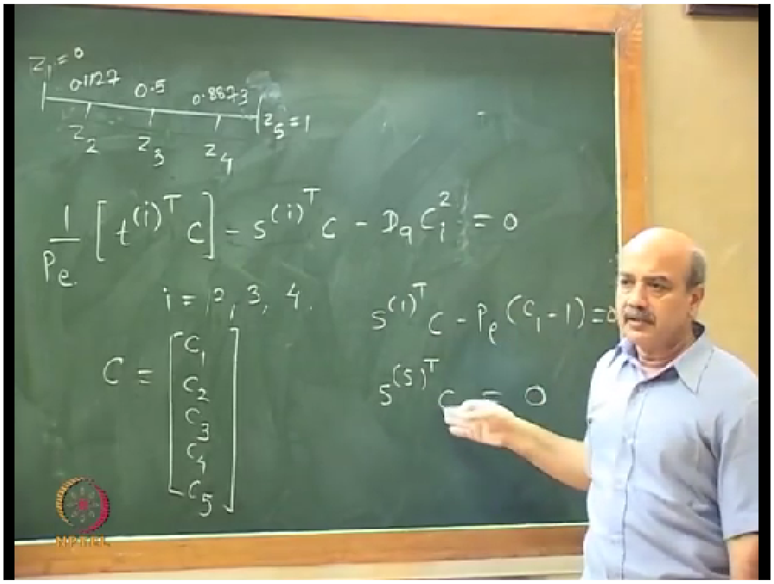
So I want to do a very simplistic solution. So this is my z1, this is my z5, okay. I have 3 collocation points, okay. This is z2 z3 z4, okay. I have taken collocation points at the roots of the third order Legandre, shifted Legandre polynomial, okay. So this happen to be, so the first one is at 0.1127, second one is at 0.5, third one is at 0.8873, okay. This z5=1 and z1=0. So we have 5 collocation points, 3 internal collocation points, 2 boundary points, okay and then you are going to get 5 equations in 5 unknowns, okay.

We are going to get 5 equations in 5 unknowns, okay. What is the first thing to do here. First thing to do is to compute A matrix, okay. First thing to do is to compute A matrix. So A matrix, okay, if I actually compute A matrix, okay, if I compute using these 4 points, it will be 5+5 matrix, okay. Then if I will compute S and T matrices, okay. I will just write here the sample of S and T matrix.

So S matrix first row comes out to be -13 14.79 -2.67 1.88 -1 and so on, okay. So this is a 5+5 matrix. So there are 5 rows, this is the last row, this is the first row, okay. Likewise, knowing these points, once I have chosen these points, I can compute A matrix, okay, then I can compute C and D matrices, then I can compute S=C*A inverse. I can compute T=D*A inverse, because A, C and D, these matrices depend only on the values of the collocation points, okay.

So once I have got these matrices, okay, then I am known to use these rows of these 2 matrices to convert this differential equation into set of algebraic equations, okay. So what is my first equation. So I have 3 equations at the 3 internal collocation points coming from this differential equation. I have 2 equations coming from boundary conditions, okay. So my equation would look something like this, okay.

(Refer Slide Time: 43:46)



So 1/Pe*ti transpose*C vector-Si transpose*C vector-DaCi square=0, i going from 2 3 4, okay. What is this C vector? C vector now consists of C1 C2 C3 C4 and C5, okay. C vector consists of C1 C2 C3 C4 C5, okay. So this row times C1 C2 C3 C4 C5, okay plus this row times C1 C2 C3 C4 C5 plus C2 square when i=2, okay. Second row, here you will choose the second row from the matrices.

Here when you chose i=3, you will choose the third row from S and T matrices, okay and you will get here Da*C3 square, okay. So you are getting 3 non-linear equations. You are getting 3 non-linear equations and the 2 additional equations arise from boundary conditions, okay. So there are 2 additional conditions that you get is S1 transpose C-Pe*C1-1=0 and S5 transpose C-=0, okay.

So these 3 equations plus these 2 equations, together they form 5 non-linear algebraic equations which need to be solved simultaneously, okay because particularly these equations, everything appears in all the equations. C1 to C5 will appear in all the equations, okay. These are coupled non-linear equations. They have to be solved (()) (46:13) using Newtons method, Newton–Raphson, in some, some non-linear equations however which, may be optimization method, whatever is at hand for you to solve this.

So this is the original problem, okay which is actually defined on the domain which is non-finite dimensional. See the true solution here, let us come back here. What is the true solution here? True solution here is the concentration profile as a function of z, okay. Z varying from 0-1. So it is a function, okay. It belongs to, which set does it belong to? It belongs to the set of continuous functions twice differentiable defined on domain 0-1.

The true solution is actually that. We have discretized the problem, okay using interpolation polynomial. When we convert it into fifth order, fifth, not fifth order, sorry, 5 dimensional vector, okay. So we are approximating in infinite dimensional solution using the fifth order, sorry 5 dimensional vector. Well, you can increase the number of collocation points to 7 8 9 but you know how many you can go.

So if you want to really make a higher dimensional approximation, what one could do is, one could divide this into segments and on each segment, one can define a lower order collocation polynomial, okay. Then of course that is called as orthogonal collocation of finite elements. Then you have to write conditions by which the neighbouring solutions meet each other. So those conditions will have to be written, okay.

“Professor - student conversation starts” (()) (48:02) Additional conditions will come.to maintain the continuity of the solution. You need additional conditions to be imposed on, okay. **“Professor - student conversation ends”** But this is the basic principle. Once you understand this, you may be extending it to finite element is not difficult. This concept is this, okay.

Now before we close this lecture, I also want to show that this is not just converting the boundary value problem. I am going to just take a version of the same problem which is the partial differential equation, okay and then you will see that the partial differential equation, okay, will get converted into an ordinary differential equation, set of ordinary differential equations.

Here there is no time involved here, okay. I am going to now convert this into a partial differential equation by including the time delivery. If I include the time delivery, the same problem, okay… Instead of making it transformed into set of coupled algebraic equations, non-linear algebraic equations, it will get transformed into set of coupled ordinary differential equations, okay.

Then of course you will use methods to solve the ordinary differential equations, that is separate thing, okay. Right now we are just looking at the problem transformations, okay. So let us do a quick recap. What we have done is, we have this second order ordinary differential equation, okay. We wanted to, we have proposed a polynomial interpolation based solution, approximate solution for this dependent variable in terms of independent variable z, okay, nth order polynomial, interpolation polynomial and then we are forcing this residual, we call this as a residual to be 0 at finite number of collocation points.

This collocation points are chosen at roots of the shifted internal polynomial, okay. Why shifted internal polynomial, why not, why not at some say regular intervals and so on. It has been found that if you actually place them and shifted them near a polynomial then the approximation errors are known, okay. So the reason for choosing orthonormal or orthogonal polynomial, roots of the orthogonal polynomial is to get a less approximation errors, okay.

So there is a reason why we choose the collocation points in the special way and nor did we say and so on. So let us not get into that part but just accept this now. If you put them at special locations, then the approximation errors are low, okay. So then we looked at one problem which is Tubular Reactor with Axial Mixing and this problem, what has happened is, we are able to convert this particular problem into set of 5 coupled non-linear algebraic equations which need to be solved interactively further, okay, which will give you an approximate solution, okay.

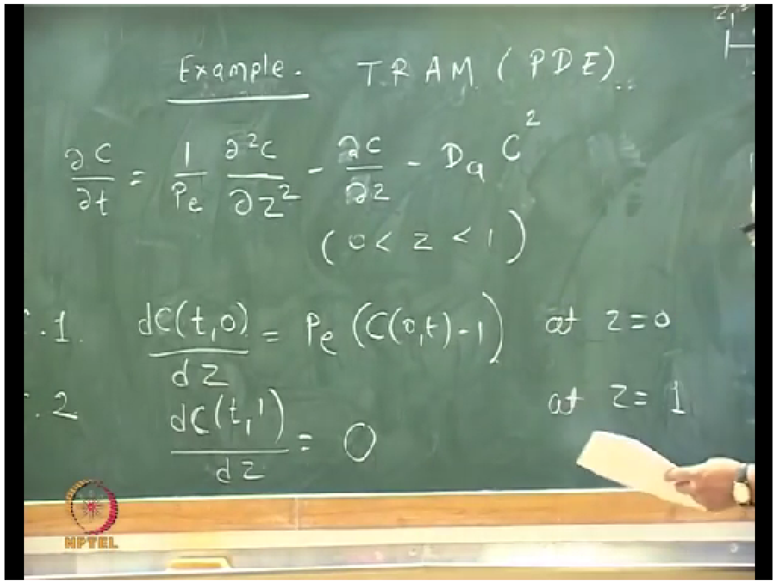
Now here because the derivative approximation is much better, okay. The derivative approximation is much better, typically it is found that a good solution can be obtained using less number of collocation points. So till now we have looked at finite difference method, okay. In the finite difference method, you will need large number of collocation points, sorry you need large number of grid points to get a good solution because you are taking local approximation of the derivative, okay, relative local approximation of the derivative.

So you need large number of grid points to get a good solution. So what is the meaning of large number of grid points? Large number of grid points means, see suppose you may not enforce this equation at this residual to be equal to 0 at large number of grid points, the number of equations that you need to solve simultaneously will be larger. Suppose to get a good solution using finite difference, I need to sub-divide this into 100 small letters.

So then here when you transform this into algebraic equations, you get 100 algebraic equations plus 2 algebraic equations at the boundary, so 102 algebraic equations, okay. What is formed here is that that this approximation less number of collocation points or smaller order polynomial gives you a good solution in many cases, okay. So here using this method, you can get good approximations using less computations, okay. So just to compare…, okay.

Let me before we just close the lecture, let us just look at the partial differential equation and then let us see what happens. Now I want to keep the same problem, okay. I am not going to change the problem except in this case, I looked at the steady state solution, I did not involve, I did not consider times but suppose you were to consider the transient response of the Tubular Reactor with Axial Mixing. Let us keep the same problem, okay.

(Refer Slide Time: 54:44)



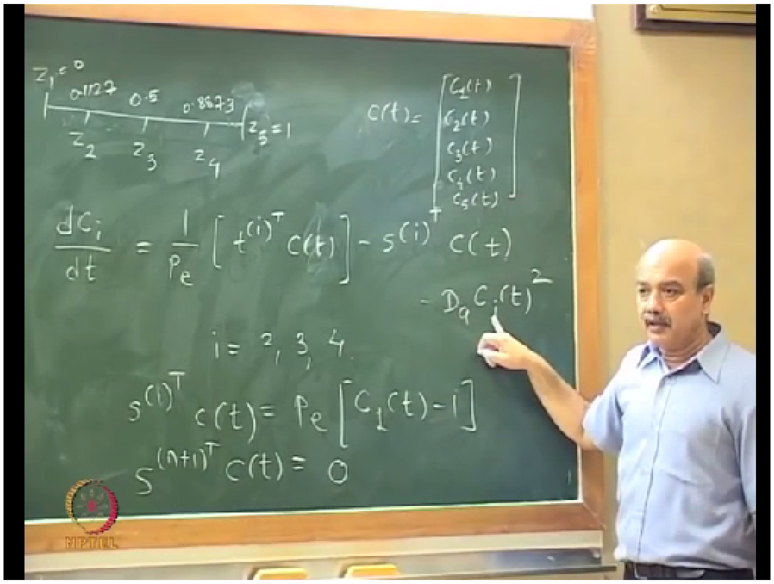
So I am going to say here do C/do t=, so these become partial derivatives, do 2C/do C square do C/do z*, so the time derivatives, earlier we had put it equal to 0, now the time derivative is not equal to 0. So now my second example that is this is PDE, okay. So I want these conditions to hold at what time. So I want this condition to hold at all the times. So dCt/dz, okay=… okay. Now when time has come into picture, okay.

And I want the solution to obey these equations at all the times, okay and all the times, so this is Ct, so now there are 2 attributes to the solution, time and space. There are 2 attributes to the solution. This is the partial differential equation. Earlier we were looking at the boundary value problem, we had only 1 attribute that is 1 independent variable that was space. Now my solution will be time and space, okay.

So when I convert this problem, when I discretize this problem, I am going to only consider and discretization in space. I am not going to write out discretize in time. I am going to keep time intact, okay and discretize only the right hand side, the spatial part, okay. So what happens if I, so at the internal collocation points, I get 3 differential equations. What are those 3 differential equations? Now there are 3 differential equations in time, okay.

The right hand side becomes algebraic, okay, because these derivatives are approximated using method of collocations, okay. This we do not have to approximate, okay, right now.

(Refer Slide Time: 57:05)



So we have this equation which is written at the 3 internal grid points, okay. dCi/dt, okay=1/Peclet number*Ti transpose C. So here now of course Ct-Si transpose Ct-dACit square, okay, i going from 2, 3 and 4, okay and then this last 2 equations, actually last 2 equations becomes 2 algebraic constraints. So what you get, what you get is set of differential algebraic system, okay.

What you get now, because these are spatial derivatives. This is the time derivative, okay. The spatial derivative will get converted into algebraic equations, okay. There are algebraic constraints to be obeyed at the boundary points, okay and this partial differential equation gets converted into ordinary differential equations, okay. So we have 3 ordinary differential equations and there are 2 algebraic constraints in the boundary that is S1 transpose Ct=…, okay.

So there are 3 differential equations, these are 3 coupled non-linear differential equations and they are coupled tightly with these 2 algebraic constraints and they have to be solved simultaneously. You can see here, what is the Ct vector? Ct vector is C1t C2t C3t C4t and C5t, okay. So dC2/dt, dC3/dt, dC4/dt, all of them are functions of C1, C2, C3, C4, C5. Not only that because of this square coming here, these are non-linear functions, okay.

So there are 3 non-linear differential equations, 2 algebraic equations coupled, okay. In this case, you may be able to eliminate 2 variables. Let us see you decide to eliminate the first and the last, okay. You decide to eliminate C1 and C5, it may be possible because these 2 are linear constraints, these 2 are linear constraints, okay. There is no non-linear integer. So you might able to rearrange and express, okay, C2, then we are able to express C1 and C5 in terms of C2 C3 C4.

If you do that, then you can eliminate here and then you will get 3 differential equations in 3 unknowns and then you can solve them by whatever method but then, that is 1 way or you solve it simultaneously using a method for solving differential algebraic systems, okay. So this is how you have transformed a problem which is originally ordinary differential equations, boundary value problem, okay, into set of algebraic equations, linear or non-linear, okay.

If it happens to be a partial differential equation, you will get a set of ordinary differential equations plus algebraic conditions, these have to be solved simultaneously and then you are able to solution, okay. So what we have learnt in this part is that how interpolation polynomials can be used to transform a problem, okay. So we began by, see what is the foundation of all this? The foundation is that, why we could construct a polynomial solution?

Because sometime back, I talked about Weierstrass theorem what does Weierstrass theorem tell you. Weierstrass theorem tells you that any continuous function can be approximated arbitrarily by using a suitable ordinary polynomial, okay. That is why we could construct a polynomial approximation with a solution Cz or Ctz, okay. That is why we could construct a polynomial approximation, okay.

Now using the polynomial approximation, how we have constructed a polynomial approximation. Weierstrass theorem there exists a polynomial function. Here you have to actually construct it. The first method that we saw was Taylor series approximation, okay that led to finite difference method. The second method that we saw was interpolation that has given rise to orthogonal collocations, okay.

In the next class onwards, we will start looking at least squares method, so least squares fitting okay and then in the context of least squares fitting is a very, very vast area and I will be talking not just about converting boundary value problems or partial differential equations, I will be talking about many more things under least squares. Now there in the context of partial differential equations or boundary value problems, we will get the method of finite element, the so called method of finite element, okay. So with this, we will close this lecture and move out to least squares methods in the next class.