Indian Institute of Science
Bangalore

NP-TEL
National Programme on
Technology Enhanced Learning

<u>Course Title</u>

Finite element method for structural dynamic And stability analyses

Lecture – 13
Mathematical preliminaries and terminologies; Euler's forward and backward difference methods.

By
Prof. CS Manohar
Professor
Department of Civil Engineering

Indian Institute of Science,
Bangalore-560 012
India

Finite element method for structural dynamic and stability analyses

Module-5

Time integration of equation of motion

Lecture-13: Mathematical preliminaries and terminologies; Euler's forward and backward difference methods

Prof C S Manohar
Department of Civil Engineering
IISc, Bangalore 560 012 India

NPTEL

1

We'll begin discussion on a new topic today, this is on time integration of equation of motion,

Numerical integration of equations of equillibrium

Consider a N-dof system

$$M\ddot{U} + C\dot{U} + KU + R\left[U(t), \dot{U}(t), t\right] = F(t)$$

$$U(0) = U_0; \dot{U}(0) = \dot{U}_0$$

Remarks:

- This equation constitutes a set of semi-discretized system of coupled second order ode-s. That is, these equations have been obtained after discretizing the spatial variables.
- This set of equations constitutes a set of initial value problems.

NPTEL

so at the end of our previous discussions we have seen that the governing equation of motion typically is formulated in this form MU double dot + U dot + KU, for sake of generality I am now introducing a nonlinear term we have not discussed how this originates, but we can believe that this type of terms would arise if we include nonlinear behavior either in strain displacement relations or in stress strain relations, so this is the equation of motion, and these are the specified initial condition, so this equation constitutes a set of coupled second-order nonlinear ordinary differential equations, we say that these equations are semi discretized equations because in arriving at this equation we have discretized space but the time is still a continuous parameter, so they also constitute a set of initial value problem, so these are a set of ordinary differential equations, semi discretized ordinary differential equations which constitute an initial value problem and these equations are coupled.

• The time variable, t, however, is still continuous.
We consider now the problem of discretizing in time.
We consider solution of the above equation at a set of discrete time instants $t_0 < t_1 < t_2 < \cdots < t_n < \cdots$ with $\Delta t_n = t_{n+1} - t_n$.

• The basic idea is to replace the derivatives appearing in the above equations by finite difference approximations and then solve the resulting algebraic equations.

NPTEL

Now we consider the problem of discretizing in time so what we aim to do is to we consider solution of this equation at a set of discrete time instants ordered as T naught, T1, T2, TN with delta TN being the step size, TN + 1 – TN. The basic idea here is to replace the derivatives appearing in the equation of motion by finite difference approximations and then solve the resulting algebraic equation that is a main idea. Now to be able to do that what we will do is we

Equations in standard form for a N-dof system

$$M\ddot{U} + C\dot{U} + KU + R\left[U(t),\dot{U}(t),t\right] = F(t); U(t_0) = U_0, \dot{U}(t_0) = \dot{U}_0$$

$$\Rightarrow \ddot{U} + M^{-1}C\dot{U} + M^{-1}KU + M^{-1}R\left[U(t),\dot{U}(t),t\right] = M^{-1}F(t)$$

Define $X_I(t) = U(t)$ & $X_{II}(t) = \dot{U}(t)$ and

$$x(t) = \begin{Bmatrix} X_I(t) \\ X_{II}(t) \end{Bmatrix} = \text{system state vector}$$

$\Rightarrow$

$$\dot{X}_I = X_{II}$$

$$\dot{X}_{II} = M^{-1}F(t) - M^{-1}CX_{II} - M^{-1}KX_I - M^{-1}R\left[X_I, X_{II}, t\right]$$

$$\begin{Bmatrix} \dot{X}_I(t) \\ \dot{X}_{II}(t) \end{Bmatrix} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix} \begin{Bmatrix} X_I(t) \\ X_{II}(t) \end{Bmatrix} + \begin{Bmatrix} 0 \\ -M^{-1}R\left[X_I, X_{II}, t\right] \end{Bmatrix} + \begin{Bmatrix} 0 \\ M^{-1}F(t) \end{Bmatrix}$$

$\Rightarrow$ Equation in state space form given by

$[x(t),t]; x(t_0) = x_0; x(t)$ is $2N \times 1$.

4

will reorganize the equation of motion in a slightly different form, so this is the equation of motion that we have MU double dot + CU dot + KU + nonlinear terms is F(t) and specified initial conditions. Now I pre multiply by M inverse so this equation now becomes, takes this form. I will introduce now a new set of variables X1 and X2, X1 is U(t), X2 is U dot(t), I can define another 2N cross 1 vector in which I assemble X1 and X2 as shown here, we call this as system state vector, so this is a state vector consisting of displacements and velocities at the end degrees of freedom.

Now what is X1 dot? X1 dot is U dot, U dot is X2, so this is X1 dot is X2, X2 dot is U double dot so that I obtained from the governing equation and we write in this form. Now I will assemble this equation in the matrix form, I will have X1 dot, X2 dot is equal to this matrix into X1, X2 plus the non-linear term, plus the excitation term. Now I can rewrite this set of equations in a general form as X dot is some function A, X(t,t) where A, this vector A is 2N cross 1, we say that this equation is in state space form, by state I mean a displacement and velocity vector together they had constitute the state, and this equation is said to be written in the configuration space where we have only displacements and we get second order differential equations, in state space we have a set of 2N first order coupled ordinary differential equations which are initial value problems and we say that the equation is in the state space form.

The set of N coupled 2^{nd} order ODE-s

$$M\ddot{U} + C\dot{U} + KU + R\left[U(t), \dot{U}(t), t\right] = F(t); U(t_0) = U_0, \dot{U}(t_0) = \dot{U}_0$$

have been thus recast as the following set of $2N$ first order coupled ODE-s

$$\dot{x} = a\left[x(t), t\right]; x(t_0) = x_0; x(t) \text{ is } 2N \times 1.$$

Denote by $x(t, x_0, t_0)$ = solution of $\dot{x} = a\left[x(t), t\right]$ with $x(t_0) = x_0$

Discrete time approximation

time: $t_0 < t_1 < t_2 < \cdots < t_n < \cdots; \Delta t_n = t_{n-1} - t_n$

Denote $y_n = x(t_n, x_0, t_0)$

How to approximate $\dot{x}$ in terms of y_n?

5

So what we have done is the set of N coupled second order ordinary differential equations given by this in the configuration space have been thus recast as the following set of 2N first order coupled ordinary differential equation. Now what I do is I introduce a notation X(t,x naught, t naught) I denote solution of this differential equation with X(t naught) = X naught, so there is T naught, X naught, and T of course is the independent variable. In discrete time approximation, time will be discretized as increasing sequence T naught, T1, T2, TN with steps as delta TN, as I mentioned before and I denote the value of the system state at T = TN by YN, so YN is X(TN) where initial conditions at T = T naught is X naught. Now the main question is how to approximate X dot in terms of YN, okay, that is the problem.

Use finite difference approximations. For example,

$$\dot{x}(t) \approx \frac{x(t+\Delta t)-x(t)}{\Delta t} \Rightarrow \dot{y}_n \approx \frac{y_{n+1}-y_n}{\Delta t_n}$$

$$\Rightarrow y_{n+1} \approx y_n + \Delta t_n a\left[y_n, t_n\right]; n = 0,1,2,\cdots \text{ with } y_0 = x_0$$

$$\text{and } \dot{y}_n = a\left[y_n, t_n\right] \quad \left(\text{given that } \dot{x}(t) = a\left[x(t), t\right]\right)$$

Here $\Delta t_n; n = 0,1,2,\cdots$ = algorthmic parameters.

Accuracy depends upon the choice of $\Delta t_n; n = 0,1,2,\cdots$

NPTEL

Now we could use finite difference approximation for example X dot(t) can be written using forward difference scheme, X dot(t) is approximately equal to X(t) + delta T – X(t)/delta T, so consequently I can write this as YN dot = YN + 1 - YN / delta TN, so from this I can derive now the equation for YN + 1, which will be YN + delta TN into YN dot, YN is nothing but A(YN,TN), where N runs from 0, 1, 2, 3, with Y naught = X naught at T = T naught. Now delta TN are the algorithmic parameter delta T1, delta T2, delta T3 are the algorithmic parameters, if all of them are equal then delta T is the only algorithmic parameter. Now the accuracy of this solution depends upon this choice of these algorithmic parameters delta TN, if delta TN are all equal the accuracy depends basically on the choice of delta T.

Errors

Local discretization error: $l_{n+1} = x\left(t_{n+1}, x_n, t_n\right) - y_{n+1}$

Global discretization error: $e_{n+1} = x\left(t_{n+1}, x_0, t_0\right) - y_{n+1}$

Round-off error: R_{n+1} (due to the use of finite precision calculations)

Dynamical system modeling errors: does the algorithm correctly capture dynamical properties of systems (such as natural frequencies, FRF, free vibration amplitudes, energy concservation,...)?

NPTEL

7

Now we talk about errors, we call something on a local discretization error, this is X(TN+1, XN, TN) – YN+1 that means this is, I have moved from T = TN to T = TN+1 to get X(TN+1) and that is approximated as YN + 1, so this error is called local discretization scheme, that means we are basically moving from TN to TN+1. On the other hand if you move from T naught to TN+1 that means all the preceding steps up to N+1 step if you cover then we call this error as the global discretization error, there is other error called round off error which is due to finite precision calculations, so you do either double precision or single precision calculation, typically we do double precision calculations, so on a computer the digits get terminated so it is inevitable so there will be errors, that we call as RN+1. Of course there are other errors which we call as dynamical system modeling errors, for example does algorithm correctly capture dynamical properties of system like natural frequencies, frequency response functions, free vibration amplitudes, is the energy conserved suppose I am dealing with an undamped free vibration problem, is the energy conserved? So if there is any compromise on any of these issues, these can be grouped as dynamical system modeling errors.

How to formulate the integration algorithms to achieve acceptable accuracy? How to understand the nature of the errors?

$$\lim_{n\to\infty}\|e_n\| \to \begin{cases} 0? \\ c<\infty? \\ \infty? \end{cases}$$

How to select algorithmic parameters to achieve acceptable performance of the integrators?

What are the expectations from a good integrator?

NPTEL

Now the basic question that we are looking for is how to formulate the integration algorithms to achieve acceptable accuracy, there's so many errors of different kinds so associated with that there is a definition of an accuracy so how to understand the nature of different errors, okay, and does this error as N tends to infinity norm of EN, does it go to 0, or does it remain finite, or does it become unbounded, so at every time step certain error is made either because of truncating of, we are approximating the derivative by a finite difference approximation and there is a round off error, so consequently at every time step there will be an error, so how does this error committed at one step, how does it propagate to the future steps, so does it go to 0 or it remains bounded or does it become unbounded. So now how to select algorithmic parameters to achieve acceptable performance of the integrators, so what are the expectations from a good integrator, so these are the type of questions we want to now address.

Dynamical system modeling errors:

Consider scalar equation $\ddot{u} + \omega^2 u = 0; u(0) = u_0, \dot{u}(0) = \dot{u}_0$

Exact solution

$$u(t) = u_0 \cos \omega t + \frac{\dot{u}_0}{\omega} \sin \omega t = R \cos(\omega t - \theta)$$

$$R = \sqrt{u_0^2 + \left(\frac{\dot{u}_0}{\omega}\right)^2} \;\&\; \tan \theta = \frac{\dot{u}_0}{\omega u_0}$$

In the discrete approximation:

- Do we get amplitude of response to be constant?
  Otherwise we get numerical dissipation or growth.
- Do we get the impulse response function correctly?
- Is the distortion acceptable?
- Is the vibration energy conserved?

Now let's spend some time on dynamical system modeling errors so that we understand what exactly is meant, so now let us consider the scalar equation of motion U double dot + omega square U = 0, so it is a single degree freedom system, natural frequency is omega and it starts with initial condition U naught and U naught dot, we can solve this problem exactly and we know that this is the solution, we have an amplitude and FS that they are given by this, they are functions of initial conditions and system natural frequency. Now the question is suppose if I solve this equation using the discrete approximation, do we get amplitude of response to be constant, for example do we get the solution in this form where R is a constant, otherwise we get numerical dissipation or growth, if amplitude is not constant either it will decay or it will grow, so then that would mean that the algorithm has introduced some artificial dissipation mechanism into the system which is not there in the original equation of motion, then do we get the impulse response function correctly, is the distribution distortions acceptable, okay, what is acceptable distortion? Now is the vibration energy conserved, some of these questions are important, so answering these questions would enable us to discuss the dynamical system modeling errors.

Dynamical system modeling errors (continued):

Similarly, for the system

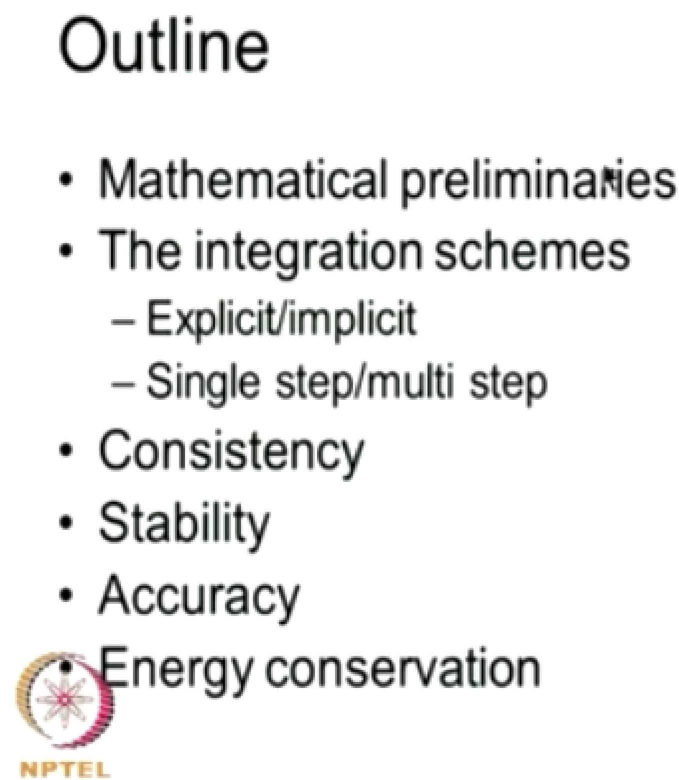
$$\ddot{u} + 2\eta\omega\dot{u} + \omega^2 u = P\exp\left(i\Omega t\right); u(0) = u_0, \dot{u}\left(0\right) = \dot{u}$$

are the properties of the frequency response curve preserved?

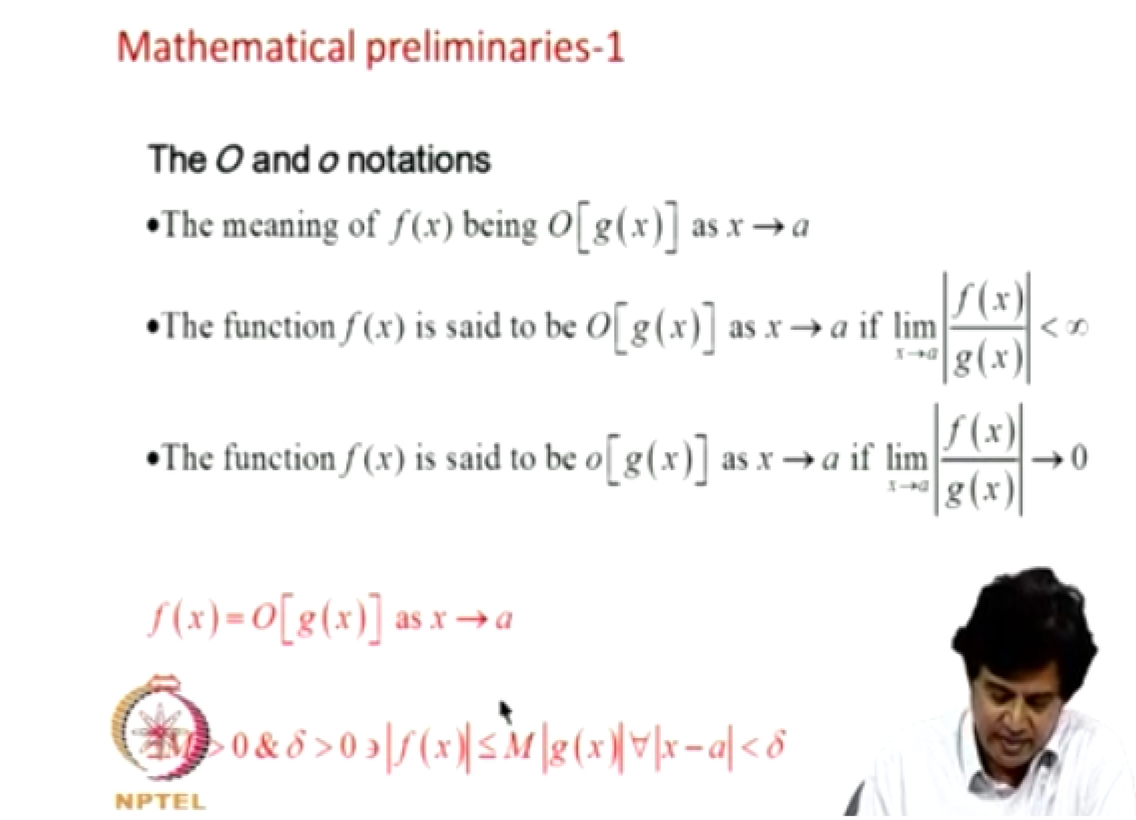
Similarly, for nonlinear systems we can pose questions on bifurcation characteristics.

NPTEL

We could also consider for example a damped harmonically driven oscillator and we know that in steady state this type of systems have certain well-known features, there is a resonance and where the resonance occurs, what is resonance amplitude, now what is the shape of the frequency response function so on and so forth, there are well known qualitative features associated with the response of the system. Now the question we can ask is does the discretized version of the solution process these well-known features or not, okay, so that is one. Similarly if there is a nonlinear term we are not discussed about nonlinear term, but we can think of adding say for example alpha U cube here, now such systems display a pattern of bifurcations we will come to that later but at this stage it is suffice to observe that nonlinear systems have certain qualitative features in the associated with their behavior, and we can ask the question does the discretized version of the equation display the same type of qualitative behavior as the continuous version, okay, this type of issues can be grouped under the heading of dynamical system modeling errors.



So what we are going to do is, there will be a few mathematical preliminaries that are needed to understand the questions and the answers that we will be discussing related to the properties of these integration schemes, and there are some set of terminologies we talked about implicit schemes, explicit scheme, single step method, multi-step methods, self-starting, and not self-starting, we talk about consistency of this, stability of these methods and we discussed inner accuracy, energy conservation properties, and so on and so forth, so what we will do is we will try to discuss these issues as we go along, so we'll start with some simple mathematical



preliminaries, we discussed this earlier in one of the lectures, we will be using this so-called gauge notations the O, and capital O and lowercase o notations, we say that F(x) is of order G(x) as X tends to A, if as X tends to A this ratio F(x)/G(x) it remains finite, okay, and if this goes to 0 then we say that it is lowercase order G(x) okay, the mathematical description is given here I will not get into these details I have provided it for sake of completion, but I will list it with few examples. Now let's consider this function A into X to the power of 7, BX cube + CX + D, we make a statement that this function is order X to the power of 7, as X tends to infinity, so how do you verify, you would divide this by X to the power of 7, okay and you can see that this goes to as X tends to infinity, this goes to A, right the first term will be A, second term will be B/X to the power of 4, C/X to the power of 6, D/X to the power of 7, so in the denominator as X tends to infinity all the terms go to 0 except the first term which is A, so we say that this statement is verified by checking this, similarly this function is order X to the power of 0 as X goes to 0, how do I check? You take extra 0, as X goes to 0 only D will remain, so this is finite, right, so this polynomial is order X to the power of 0.

Examples

- $ax^7 + bx^3 + cx + d$ is $O\left(x^7\right)$ as $x \to \infty \because \lim_{x \to \infty} \left| \frac{ax^7 + bx^3 + cx + d}{x^7} \right| \to a < \infty$
- $ax^7 + bx^3 + cx + d$ is $O\left(x^0\right)$ as $x \to 0 \because \lim_{x \to 0} \left| \frac{ax^7 + bx^3 + cx + d}{x^0} \right| \to d < \infty$
- ax^7 is $O\left(x^7\right)$ as $x \to 0 \because \lim_{x \to 0} \left| \frac{ax^7}{x^7} \right| \to a < \infty$
- ax^7 is not $O\left(x^8\right)$ as $x \to 0 \because \lim_{x \to 0} \left| \frac{ax^7}{x^8} \right| \to \infty$

NPTEL

Now A to the power of, X to the power of, A into X to the power of 7 is order X to the power of 7 as X goes to 0, how do you verify, you divide by X to the power of 7 and go take X to power of 0, I get A, which is finite, so there are a few more examples you can verify, you know, get a feel for what these terminologies mean. So I have a few more examples, let’s quickly run

Examples

- $\sin(x)$ is $O(x)$ as $x \to 0 \because \lim_{x\to 0}\left|\frac{\sin(x)}{x}\right| \to 1 < \infty$
- $\sin(x^2)$ is $O(x^2)$ as $x \to 0 \because \lim_{x\to 0}\left|\frac{\sin(x^2)}{x^2}\right| \to 1 < \infty$
- $\cos(x)$ is $O(x^0)$ as $x \to 0 \because \lim_{x\to 0}\left|\frac{\cos(x)}{x^0}\right| \to 1 < \infty$
- $\sin(x)$ is $o(x^0) = o(1)$ as $x \to 0 \because \lim_{x\to 0}\left|\frac{\sin(x)}{x^0}\right| \to 0$
- $\cos(x)$ is $O\left(x^{-\frac{1}{2}}\right)$ as $x \to 0 \because \lim_{x\to 0}\left|\frac{\cos(x)}{x^{-\frac{1}{2}}}\right| = \lim_{x\to 0}\left|\sqrt{x}\cos(x)\right| \to 0 < \infty$

NPTEL

14

through this sin X is order X, as X goes to 0, why? You divide sin X/X and take X to 0 we know that this limit is 1, which is finite. Similarly sin X square is order X square by the same logic we come to this conclusion, cos X is order 1 as X goes to 0, because cos X/1, as X goes to 0 is 1, and so on and so forth, so there are few more examples I'll leave it for you to verify.

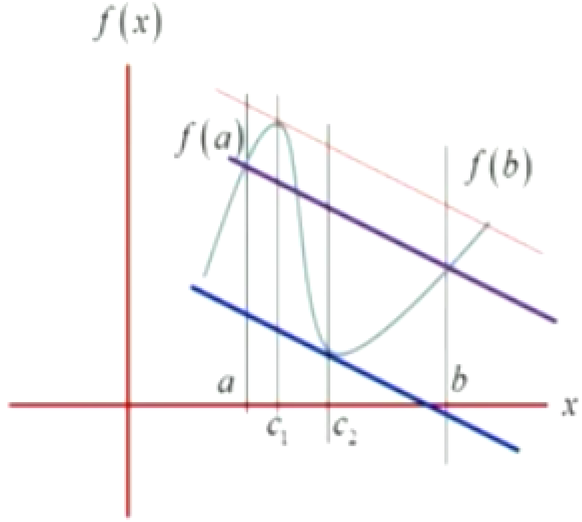
Mathematical preliminaries-2

Mean value theorem

Consider a function $f(x)$ to be continuous for $a \le x \le b$ and differentiable for $a < x < b$. According to the mean value theorem, there exists at least one c satisfying $a < c < b$ such that

$$\frac{df}{dx}(c) = \frac{f(b) - f(a)}{b - a}$$

$$\Rightarrow f(b) = f(a) + (b - a)\frac{df}{dx}(c)$$



NPTEL

15

There is one another mathematical preliminary that we would be needing, we should quickly recall what is the mean value theorem, now consider a function F(x) to be continuous for X lying between A and B, and differentiable for, this is the closed interval, this is the open interval, it is differentiable in the open interval, then according to the mean value theorem there exists at least one C satisfying the condition that C lies between A and B such that DF/DX at C is exactly given by this F(B) – F(A) / B - A, that you can see here, this is my F(x) and this is A and this is B, and you can see that at this point, and at this point, if you draw the curve DF/DX which is given by this, this will be parallel to the 2, you know this is the approximation so this will be parallel to this, which is a tangent to F(x) at those points, so at C1 and C2 this is an exact representation, but of course for a given F(x) we would not know where is that C, how many of them are there, whether there is you know where it is located we would not know.

Mathematical preliminaries-3

Taylor's series

Let f be a function of x with derivatives of all orders throughout an interval containing the point a. The Taylor series generated by f at $x = a$ is given by

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!}(x-a)^k = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \cdots$$

Now Taylor’s series is something that we are all familiar with, so let’s quickly see the statement of some of the results associated with Taylor’s series expansions, so again let F be a function of X with derivatives of all orders throughout an interval containing point A. The Taylor’s series generated by F at X = A is given by F(A) + F prime A into X – A, F double prime A by 2 factorial X - A whole square and so on and so forth, this is the well-known Taylor's series. Now

Mathematical preliminaries-3

Taylor's polynomial

Let f be a function of x with derivatives of orders $n = 0, 1, 2, \cdots, N$ throughout an interval containing the point a. Then for any $n \in [0, N]$ the Taylor polynomial of order n generated by f at $x = a$ is given by

$$P_n(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n.$$

NPTEL

if you truncate the Taylor’s series at say the nth term we get what is known as the Taylor's polynomial, so that is other conditions remaining the same I write PN(x) as this, so this is known as Taylor’s polynomial, okay.

Corollary to Taylor's theorem

If f has derivatives of all orders in an open interval I containing a, then for each positive integer n and for each x in I,

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n + R_n(x)$$

where $R_n(x) = \dfrac{f^{(n+1)}(c)}{(n+1)!}(x-a)^{n+1}$ for some c between a and x.

If $\lim_{n\to\infty} R_n(x) \to 0 \forall x \in I$, we say that the Taylor series generated by f at $x = a$ converges to f on I, and we write

$$f(x) = \sum_{n=1}^{\infty} \frac{f^{(n)}(x)}{n!}(x-a)^n$$

NPTEL

Now the corollary to the Taylor's theorem, if F has derivatives of all orders in an open interval I containing A, then for each positive integer N and for each X in the interval I, F(x) can be written as F(A) + F prime A, X - A so on and so forth after nth term there is a reminder term, and this reminder is given by this for some C between A and X, this again we are using mean value theorem in writing this, now if this limit of this remainder term goes to 0, right for all X in I, we say that Taylor series generated by F at X = A converges to F on I, and we write F(x) is this, okay.

Mathematical preliminaries-4

Forward difference approximation to a derivative

$$f(x+h)=f(x)+hf'(x)+\frac{h^2}{2!}f''(x)+\frac{h^3}{3!}f'''(x)+\cdots$$

$$\Rightarrow f'(x)=\frac{f(x+h)-f(x)}{h}-\frac{h}{2!}f''(x)-\frac{h^2}{3!}f'''(x)+\cdots$$

$$\Rightarrow f'(x)=\frac{f(x+h)-f(x)}{h}+O(h) \text{ as } h\to 0$$

Backward difference approximation to a derivative

$$f(x-h)=f(x)-hf'(x)+\frac{h^2}{2!}f''(x)-\frac{h^3}{3!}f'''(x)+\cdots$$

$$\Rightarrow f'(x)=\frac{f(x)-f(x-h)}{h}+\frac{h}{2!}f''(x)-\frac{h^2}{3!}f'''(x)+\cdots$$

$$\Rightarrow f'(x)=\frac{f(x)-f(x-h)}{h}+O(h) \text{ as } h\to 0$$

19

Now we talked about forward difference, backward difference, and central difference, where I talked about finite difference, so some of the examples are forward difference, backward difference, central difference, so let's see quickly what it means, so let's consider a function F(x) and I write the Taylor's expansion F(x) + H is F(x) + HF prime X + H square/2 factorial F double prime FX and so on and so forth. Now from this I can write, solve for F prime(x), F prime(x) will be given by F(x) + H - F (x) / H, so you are taking these, other term to the other side you are dividing by H therefore this becomes H, H/2 factorial + H square by 3 factorial and so on and so forth. So this term is clearly of the order H, because if you divide by H and take H20 this will be a finite quantity, right, so this we say that this is a forward difference approximation to F(x) at X, the backward difference approximation to derive that we consider instead of F(x) + H, I consider F(x) – H, so this will be F(x) - HF prime(x) and so on and so forth.

Again if I solve for F prime(x) I get this expression, and divided by, we have divided by H and we can see that these terms are of order H, as H goes to 0, so this approximation is known as backward difference approximation to F prime(x). To derive the central difference

Central difference approximation to a derivative

$$f(x+h)=f(x)+hf'(x)+\frac{h^2}{2!}f''(x)+\frac{h^3}{3!}f'''(x)+\cdots$$

$$f(x-h)=f(x)-hf'(x)+\frac{h^2}{2!}f''(x)-\frac{h^3}{3!}f'''(x)+\cdots$$

$$\Rightarrow f(x+h)-f(x-h)=2hf'(x)+2\frac{h^3}{3!}f'''(x)+\cdots$$

$$\Rightarrow f'(x)=\frac{f(x+h)-f(x-h)}{2h}+O(h^2)\text{ as } h\to 0$$

Intutively

Error is of $O(h)$ and we half step size $\Rightarrow$ error gets halved.

Error is of $O(h^2)$ and we half step size $\Rightarrow$ error gets quartered.

approximation what we do is we consider F(x) + H which is this expansion, and also F(x) - H which is this expansion, now I subtract these two, so if moment I subtract these 2, these 2 get cancelled, these 2 add up, I get 2H F prime(x), and similarly this H square term will get cancelled, and I will get the next term will be of the 2H cube / 3 factorial so on and so forth. Now if I solve for F prime(x) from this, I get F prime(x) of X is F(x) + H - F(x) - H / 2H + this term will be divided by H, so this will be order H square, so as H goes to 0, okay, so this is the central difference approximation for derivative of F at X. Intuitively we can see that if error is of the order H, and if we have H, that is the step size H the error gets halved, similarly if the order, error is of the order H is square and if we half the step size the error will get quartered, okay so that is the advantage of order H square, okay.

Standard form

$$M\ddot{U}(t)+C\dot{U}(t)+KU(t)+R\left[\ddot{U}(t),\dot{U}(t),U(t),t\right]=F(t)$$

$U(0),\dot{U}(0)$ specified

$$\Rightarrow \dot{x}=a\left[x(t),t\right];x(0)=x_0;0\le t\le t_f$$

Strategy: replace derivatives by finite difference approximations.

Introduce $0<t_1<t_2<\cdots<t_N=t_f$ & denote $y_n=x(t_n,0,x_0)$

Forward difference approximation

$$\text{At time}=t,\ \dot{x}(t)=\frac{x(t+\Delta t)-x(t)}{\Delta t}=a\left[x(t),t\right]$$

$$\Rightarrow y_{n+1}=y_n+\Delta t\dot{y}_n=y_n+\Delta t a(y_n)+O(\Delta t)$$

Backward difference approximation

$$\text{At time}=t+\Delta t,\ \dot{x}(t+\Delta t)=\frac{x(t+\Delta t)-x(t)}{\Delta t}=a\left[x(t+\Delta t),t+\Delta t\right]$$

$$\Rightarrow y_{n+1}=y_n+\Delta t\dot{y}_{n+1}=y_n+\Delta t a\left[y_{n+1}\right]+O(\Delta t)$$

NPTEL 21

Now let’s return to the equilibrium equation in the standard form, we already seen that this equation can be written in this standard form in the state space, now what I will do is, I will replace this X dot by finite difference approximation and see what we get, so we will consider a sequence of increasing time instant 0, T1, T2, T capital N, capital TN is TF which is a final time instant up to which we want to integrate this equation, and we denote YNS value of the system state at T = TN by starting at T = 0 with an initial condition X naught, now for X dot (t) if I use forward difference approximation, X dot(t) will be written as X(t) + D delta T - x of T / delta T, and this must be equal to A(x(t), t), so now if I solve for X(t + delta t) which is YN+1 will be equal to YN, that is X(t) + delta T is on the other side, delta T YN dot, YN dot is A(YN), so I’ll get YN+1 as YN + delta TA YN and this order of approximation is order delta T, okay.

Now if we want now backward difference approximation we consider time instant T + delta T, and go back in time and I get X dot(t) + delta T is X(t) + delta T – X(t) / delta T, so this looks similar to this but here you must notice that I am writing this at time instant T, okay, T + delta T is, I heard of the time at which I am writing this whereas here the T is lagging, okay T + delta T is the current time and whereas T is the previous time instant. Now here again I can write YN + 1 is YN + delta T this is important, this is Y dot(N+1) whereas here it is Y dot(n), so this will be YN + delta T, A(YN+1), whereas this is A(YN) and the order of accuracy is still order of delta T, the central difference scheme I can write this so again at time T, I write X dot(t) is

Central difference approximation

$$\text{At time}=t,\ \dot{x}(t)=\frac{x(t+\Delta t)-x(t-\Delta t)}{2\Delta t}=a\left[x(t),t\right]$$

$$\Rightarrow y_{n+1}=y_{n-1}+2\Delta t\dot{y}_n=y_{n-1}+2\Delta t a(y_n)+O(\Delta t^2)$$

Trapezoidal rule

$$x(t)=\int_0^t \dot{x}(s)ds \Rightarrow x(t_{n+1})=\int_0^{t_{n+1}} \dot{x}(s)ds=\int_0^{t_n} \dot{x}(s)ds+\int_{t_n}^{t_{n+1}} \dot{x}(s)ds$$

$$=y_n+0.5\Delta t(\dot{y}_{n+1}+\dot{y}_n)$$

$$\Rightarrow y_{n+1}=y_n+0.5\Delta t(\dot{y}_{n+1}+\dot{y}_n)$$

$$=y_n+0.5\Delta t\{y_{n+1}+\Delta t a(y_{n+1})+y_n+\Delta t a(y_n)\}$$

NPTEL

22

X(T+delta T) – X(t) – delta T/2 delta T, this must be equal to A(x(t),t), so I want to write for X(T+delta T) I will get YN + 1, this is YN – 1, 2 delta T into YN dot, this is YN dot, so that is YN – 1 + 2 delta TA(YN) and this approximation is order delta T square, okay.

Now we can also develop another scheme which is slightly different from, logic is slightly different suppose X(t) I write it as 0 to T, X dot (s) DS, that's a definition of a derivative, so if you consider T = TN+1, this will be 0 to TN+1, X dot(s) DS. This 0 to TN+1 I can write it as 0 to TN + TN to TN+1, so this first term is nothing but X(TN) and this is the increment. For this second term I will use the trapezoidal rule of integration, so I will therefore I will get this as YN+1 as YN + 1/2 of delta T, YN dot N+1 +YN dot , okay, so from this I get this as the approximation. Now for YN + 1, Y dot N+1, I will write A(YN+1), and YN dot I'll write it as YN + delta TA YN, so this approximation is known as the approximation based on trapezoidal rule.

Forward difference approximation

$$y_{n+1} = y_n + \Delta t \dot{y}_n = y_n + \Delta t a(y_n) + O(\Delta t)$$

Backward difference approximation

$$y_{n+1} = y_n + \Delta t \dot{y}_{n+1} = y_n + \Delta t a[y_{n+1}] + O(\Delta t)$$

Central difference approximation

$$y_{n+1} = y_{n-1} + 2\Delta t \dot{y}_n = y_{n-1} + 2\Delta t a(y_n) + O(\Delta t^2)$$

Trapezoidal rule

$$y_{n+1} = y_n + 0.5\Delta t \{y_{n+1} + \Delta t a(y_{n+1}) + y_n + \Delta t a(y_n)\}$$

Generic form

$$y_{n+1} = \alpha_1 y_n + \alpha_2 y_{n-1} + \cdots + \alpha_m y_{n+1-m} + \Delta t[\beta_0 \dot{y}_{n+1} + \beta_1 \dot{y}_n + \cdots + \beta_k \dot{y}_{n+1-k}]$$

NPTEL

Now we can summarize all this, so in the forward difference scheme I had this representation in backward different scheme I had this representation, in central difference I had this, in trapezoidal rule I had this. Now a generic form based on this we can see that a generic form can be written as YN+1 is alpha 1YN + alpha 2YN – 1, so on and so forth, alpha M YN+1 - M + delta T this is on Y(t) and these are derivatives, some beta naught Y dot N+1, beta 1 Y dot N, so on and so forth up to the K-th term. This is a general form into the, all these schemes will fit into these by selecting alpha 1, alpha 2, etcetera,, beta naught, beta 1 etcetera in a suitable manner I can recover these schemes. Let’s consider the generic form again, you look at the term

Generic form

$$y_{n+1} = \alpha_1 y_n + \alpha_2 y_{n-1} + \cdots + \alpha_m y_{n+1-m} + \Delta t\left[\beta_0 \dot{y}_{n+1} + \beta_1 \dot{y}_n + \cdots + \beta_k \dot{y}_{n+1-k}\right]$$

Definitions

- $\beta_0 \neq 0 \Rightarrow y_{n+1}$ depends upon derivative of the state at t_{n+1}, that is, upon $\dot{y}_{n+1}$. The scheme is said to be implicit. Other wise the scheme is said to be explicit.

- If $\alpha_2 = \alpha_3 = \cdots \alpha_m = 0, \& \ \beta_2 = \beta_3 = \cdots \beta_k = 0$, the scheme is said to be a single step scheme; otherwise it is called multi-step scheme.

- Scheme is said to be self starting if y_n for $n < 0$ does not enter the calculations.

NPTEL

beta naught, so I am writing YN + Y at T = TN+1 and this involves the derivative of the state at N + 1, if B naught is not 0, okay if B naught is not 0 therefore YN plus depends upon derivatives of the state at TN+1 that is upon Y dot N+1, so such schemes are known as implicit schemes, okay, otherwise if beta naught is 0 then the scheme is said to be explicit.

Now if we now consider alpha 2, alpha 3, and alpha M as 0, and beta 2 up to beta M are 0, then we say that the scheme is a single step scheme, otherwise it is called multi-step scheme. Now the scheme is set to be self-starting if YN for N less than 0 does not enter the calculations, okay, right, so if I write this for N this term is a previous step, okay, suppose if N = 0, this will be YN and this is alpha 1 Y naught, there is no problem, but in certain schemes you will soon see that I want to start with N+1 = 0 and this I will be needing Y of – M, and such schemes are said to be not self-starting okay we need to see that, so let’s quickly see in this scheme of things how does

Forward difference approximation

$$y_{n+1} = y_n + \Delta t\dot{y}_n = y_n + \Delta t a(y_n) + O(\Delta t)$$

Backward difference approximation

$$y_{n+1} = y_n + \Delta t\dot{y}_{n+1} = y_n + \Delta t a[y_{n+1}] + O(\Delta t)$$

Central difference approximation

$$y_{n+1} = y_{n-1} + 2\Delta t\dot{y}_n = y_{n-1} + 2\Delta t a(y_n) + O(\Delta t^2)$$

Trapezoidal rule

$$y_{n+1} = y_n + 0.5\Delta t\left\{y_{n-1} + \Delta t a(y_{n-1}) + y_n + \Delta t a(y_n)\right\}$$

Generic form

$$y_{n+1} = \alpha_1 y_n + \alpha_2 y_{n-1} + \cdots + \alpha_m y_{n+1-m} + \Delta t\left[\beta_0 \dot{y}_{n+1} + \beta_1 \dot{y}_n + \cdots + \beta_k \dot{y}_{n+1-k}\right]$$

Forward difference approximation: single step, explicit, self-starting.

Backward difference approximation: single step, implicit, self-starting.

Central difference approximation: multistep, explicit, not self-starting.

Trapezoidal rule: single step, implicit, self-starting.

this different classification schemes can be viewed as, suppose if you now consider forward difference approximation which is given by this, now you can see that this is a single step explicit self-starting scheme okay, explicit because YN plus, we have YN+1 here and there is no derivative of YN+1 on the other side, backward difference scheme is again single step implicit, it is implicit self-starting okay, because you see here YN+1 is here A(YN+1) is nothing but YN+1, derivative of YN+1 so this is implicit, backward scheme is implicit. Similarly central difference approximation is, it is a multi-step method it is explicit, and it is not a self-starting scheme because here YN - 1 you will see that in the due course, for example for N = 0, this will be YN and I will be needing Y of – 1, and Y of - 1 is a hypothetical quantity which is not there as a part of definition of the original problems of mechanics, problem of mechanics, so this is not a self-starting scheme and we need to devise some special methods to start the solution.

Similarly trapezoidal rule is single step implicit and it is self-starting, okay, why implicit, because I have A(YN+1) here which is YN+1 dot, okay so if system is non-linear you can see that in implicit schemes you need to solve a times every time step a algebraic non-linear equation okay, so the implicit schemes that way demand computational efforts, especially if system is non-linear, if system is linear we can take these terms to the left side and rearrange the terms which is simply straight forward, so the difference between implicit scheme and explicit scheme will be strongly felt in computational effort for non-linear systems.

Remarks

- $t_0 = 0 < t_1 < t_2 < \cdots < t_N = t_f$
- $t_n = n\Delta t$ [constant step size]
- System states: $U(t), \dot{U}(t), \ddot{U}(t)$
- Assume: System states at $t = t_n$ are known. To find system states at $t = t_{n+1}$.

Integration algorithms essentially achieve this.

- Computation effort is proportional to number of time steps to advance solution from $t = 0$ to $t = t_f$. The step size, therefore, need not be smaller than what is essential. We need to be concerned about
  - Growth of errors (stability) and
  - Accuracy of the solution.
  - Undamped free vibration of a sdof system:

    Check for amplitude

    Frequency distortions

Okay, let's make some observations and remarks so we have discretized time T naught to TF in terms of 0, T1, T2, TN which is a non-decreasing sequence of, actually increasing sequence of time instance, if we now take TN to be N delta T, we say that we have a constant step size, okay, now system states are U(t), U dot(t) and U double dot(t) in our calculations, now if we assume that system states at T = TN are known and that is what we assume then the problem on hand is to find system states at T = TN+1, the integration algorithms essentially achieve this, so these are known at time marching techniques, so we move from TN to TN+1 and we want to take state from TN to TN+1, so these are called time marching techniques.

Now the computation effort is proportional to the number of time steps, okay to advance the solution from T to 0 to T to TF, therefore the step size need not be smaller than what is essential they say what is the meaning of something being essential, it is something with accuracy basically, so what we need to be concerned about, we need to be concerned about growth of errors, does error committed at a time steps AT and grow as time advances, if it grows we say that the solutions are unstable, the scheme is unstable otherwise it is, if it remains constant we say it is stable, if it goes to 0 we say that it is as impractically stable, the accuracy of the solution step, if the errors do not grow, it does not mean that we get accurate answers, okay, how to judge accuracy? That is a different issue, now for example for undamped free vibration of a single degree freedom system we can check for amplitude, amplitude should remain constant and frequency distortions must not be there, we know that frequency let's say for example square root K/M do we get that square root K/M in numerical simulations or not.

Remarks (continued)

•The time integration method is the most generally applicable method to solve equillibrium equations:

For linear systems, the solution does not require transformation to natural coordinates. This enables treatment of general class of viscous damping models without stipulating damping to bee classical. On the other hand, the method requires that damping be specified in terms of the *C* matrix and not in terms of modal damping ratios. Procedures to derive *C* matirx in terms of specified modal damping ratios have been discussed in this course. The excitation could be periodic, steady state aperiodic, or transient in nature.

The method remains applicable even when governing equations are nonlinear in nature.

The time integration method is the most generally applicable method to solve equilibrium equations, we have talked about frequency response function, based methods, a dynamic stiffness, and so on and so forth, they are valid only for linear systems and of course there are additional requirements that system should reach harmonic steady-state, but here the system can be nonlinear, the excitations could be transient, this method remains applicable. Now for linear systems the solution actually does not require transformation to natural coordinates, see for linear system we can do, we can find natural frequencies, mode shapes, and uncouple the equation we have that option, but if you are trying to integrate the equation numerically it is not needed to actually perform the uncoupling of equations of motion, if you can do it, it helps but it is not essential, so this would mean that we have greater flexibility in modeling damping, see we have seen that if damping is classical there are many advantages in doing a more analysis based on mode superposition, but if you are doing direct integration there is no special requirement on what should be the damping matrix, because you can directly integrate.

Now but in practice we prefer to specify damping in terms of modal ratios, in that case damping will be specified in terms of as formulation which employs normal modes, natural coordinates and normal modes, so if you intend to use direct integration in such situations you have to formulate the C matrix from the known information about modal damping ratios, so we have discussed how to do that in the previous class. As I already said the method remains applicable even when equations of motions are nonlinear.

Remarks (continued)

For linear systems, the size of the model (dof-s) depend upon the details of mesh used in spatial discretization. These details needs to be chosen such that the system behavior is well represented over the frequency range over which excitations have significant power. If Ω_{max} is the highest frequency present int he excitations, then all the modes present in the frequency range up to about $1.25\Omega_{max}$ needs to be correctly captured. If the structure has, say, r modes in frequency range up to $1.25\Omega_{max}$, then the mesh details should be such that the model has at least $10r$ dof-s.

NPTEL

Now let's look at modeling of linear multi degree freedom systems, so how many degrees of freedom should be included in our model, we will discuss this in again at a later stage but we can start asking these questions at this stage, now the size of the model that is number of degrees of freedom actually depend on details of mesh used in the spatial discretization, now these details need to be chosen such that the system behavior is represented with acceptable accuracy over a given frequency range, what is the frequency range that we should select? We should look at excitation, suppose you are dealing with say earthquake like excitation the frequency range can be up to say about 20 to 30 Hertz from low frequency to up to that frequency, wind it could be 0 to 2 Hertz or 3 Hertz, wave also in the same region, so if you are performing say seismic response analysis of engineering structure you should ensure that say within the frequency range of say 0 to 20 Hertz whatever are the natural modes that the structure possess all of them should be modeled accurately, so we have seen that if you want to capture first say 5 modes in a model, you should have about 50 degrees of freedom in your model, that means the spatial discretization should be fine enough so that you get a model with 50 degrees of freedom so that at least one tenth of the natural frequencies will be computationally trustworthy, so that would mean if omega max is the highest frequency we will say that the discretization scheme should be such that all the modes that lie in frequency range up to 1.25 Omega max need to be captured well, so the mesh details should be such that the model has at least 10R degrees of freedom, where R is the number of degrees of freedom, the R is the number of modes that you expect the structure to have in frequency range up to 1.25 omega max, a consequence of this is that when you are using direct integration you will always

Remarks (continued)

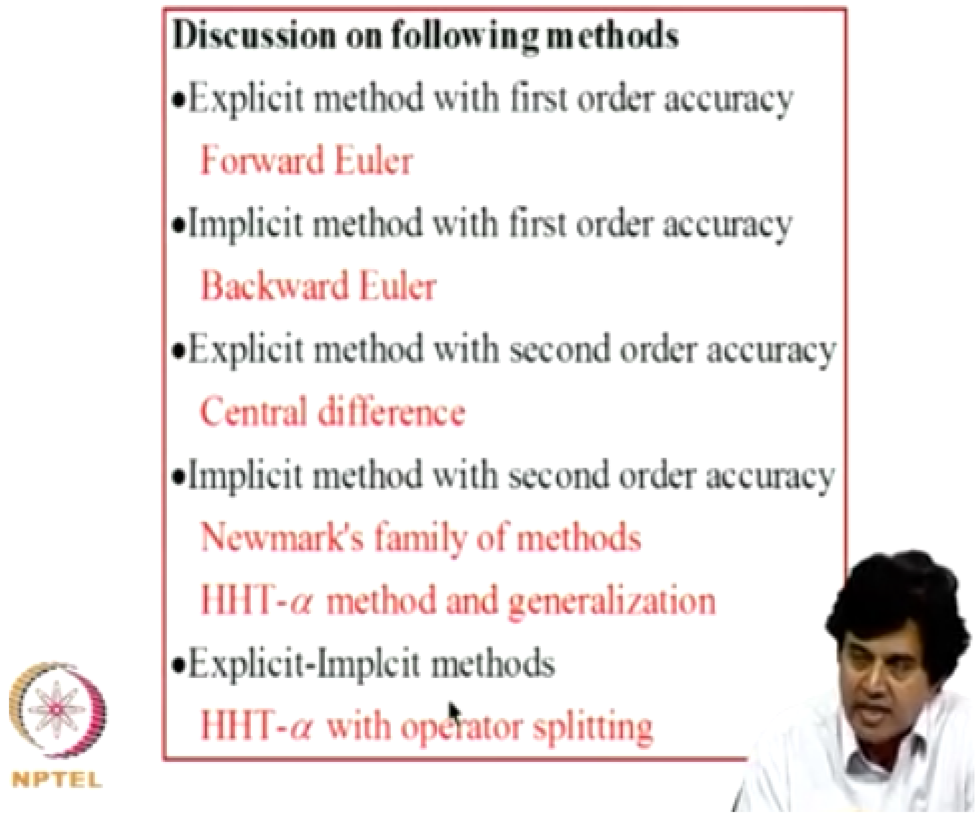
This invariably means that there would be a large number of hgiher order modes that are present in the model (with frequencies beyond Ω_{max}) which however do not contribute significantly to the response.

It is desirbale that the discretization scheme used should have inherent capability to (numerically) dissipate the higher order spurious modes and yet the same time not distort the lower order modes which contribute significantly to the response.

NPTEL

be dealing with models which have, which has spurious higher-order modes, suppose you are making a model with 100 degrees of freedom we know that the first 10 modes are likely to contribute to the response, and modes from 10 to 100 are the higher order spurious modes.

Now it is desirable that the discretization scheme that is the time discretization scheme should have some inherent capability to numerically dissipate the higher order spurious modes, but not affecting the genuine lower order modes which contributes significantly to the response, so this would mean that we look for certain numerical dissipation characteristics in our integration schemes when we formulate the time integration schemes, so what is desirable is that the lower modes should not be numerically dissipated, but the higher spurious mode should be numerically dissipated, okay, we will see more about that, but this is one of the concern that we need to appreciate at the outside.



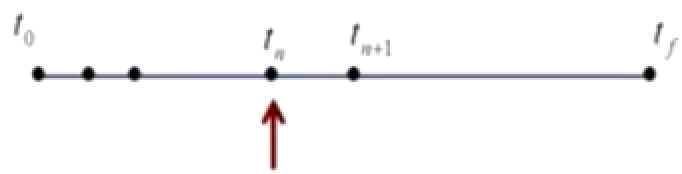
So what we will do now is will take up discussions on few methods I propose to discuss the following method, the forward Euler method, the backward Euler method, the central difference method, and there are methods known as Newmark's family of methods, and HHT alpha method, and so on and so forth, so we will see the development of these methods and as we go along we will see what motivates, what are the motivations to develop different schemes after getting, for example backward Euler scheme what is the need to go for central difference method, and what is the need to go to Newmark's family of methods when you already have done central difference method and so on and so forth. So the questions will be on stability, accuracy, dissipation of higher or spurious modes and so on and so forth, methods being implicit or explicit, these issues also will come up.

Strategy

- Development of basic formulary
- Pseudocode for implementation
- Analysis of the method
- Illustrative examples
- Discussion on relative merits

So what we, the strategy we will take is, we will develop the basic formulary for each of these schemes, and I will provide a Pseudocode for implementation, then we will analyze the method for its, how the errors behave in each, the specific method and then we will illustrate each of these methods with a few examples and we'll conclude by discussing the relative merits of different methods, so this is a scheme of things that we will follow.

Forward Euler Method



$$M\ddot{U}(t)+C\dot{U}(t)+KU(t)=F(t)$$

$$U(0)=U_0;\dot{U}(0)=\dot{U}_0;0\le t\le t_f$$

$$\dot{U}(t)=\frac{U(t+\Delta t)-U(t)}{\Delta t}\Rightarrow U_{n+1}=\Delta t\dot{U}_n+U_n$$

$$\ddot{U}(t)=\frac{\dot{U}(t+\Delta t)-\dot{U}(t)}{\Delta t}\Rightarrow \dot{U}_{n+1}=\Delta t\ddot{U}_n+\dot{U}_n$$

This is an explicit scheme

$\Rightarrow$ Condtion for equillibrium at t_{n+1} is given by

$$M\ddot{U}_{n+1}+C\left(\Delta t\ddot{U}_n+\dot{U}_n\right)+K\left(\Delta t\dot{U}_n+U_n\right)=F_n$$

$$\Rightarrow \ddot{U}_{n+1}=-M^{-1}\left[C\left(\Delta t\ddot{U}_n+\dot{U}_n\right)+K\left(\Delta t\dot{U}_n+U_n\right)-F_n\right]$$

32

So let me start with the discussion on forward Euler method, so this is a time axis T naught is here, TF is a final time instant which is here, so let’s the equation of equilibrium is MU double dot + U dot + KU = F(t) at time T, these are the initial conditions. Now at time T, I will approximate the velocity U dot(t) using forward difference scheme, which is U(t) + delta T – U(t) delta T, so from this I can get UN+ 1 which is U(t) + delta T as delta T into U dot, that is U dot N + UN, a similar approximation I can make for acceleration also, so that will be U dot(t+delta t) – U dot(t) / delta T, where U dot (N+1) is obtained as delta T, UN double dot + UN dot, so you can see here this is an explicit scheme, because when I am writing the state at N+1 I do not need acceleration at N+1, okay, now so let us consider the condition for equilibrium at TN+1, so this is written as MU double dot N+1 plus for UN dot UN+1 dot I will use this, okay this is this, then for UN+1 I will use this, so this is this, this is equal to FN where F1 is F(tn). Now I will rearrange these terms what is not known is UN+1 double dot and that is obtained in terms of the remaining terms like this.

Implementation of the forward Euler Method

(1) $n = 0; t = 0;$ Input U_0 & $\dot{U}_0$

$$\ddot{U}_0 = -M^{-1}\left[C\dot{U}_0 + KU_0 - F_0\right]$$

(2) $\ddot{U}_{n+1} = -M^{-1}\left[C\left(\Delta t\ddot{U}_n + \dot{U}_n\right) + K\left(\Delta t\dot{U}_n + U_n\right) - F_n\right]$

(3) $\dot{U}_{n+1} = \Delta t\ddot{U}_n + \dot{U}_n$

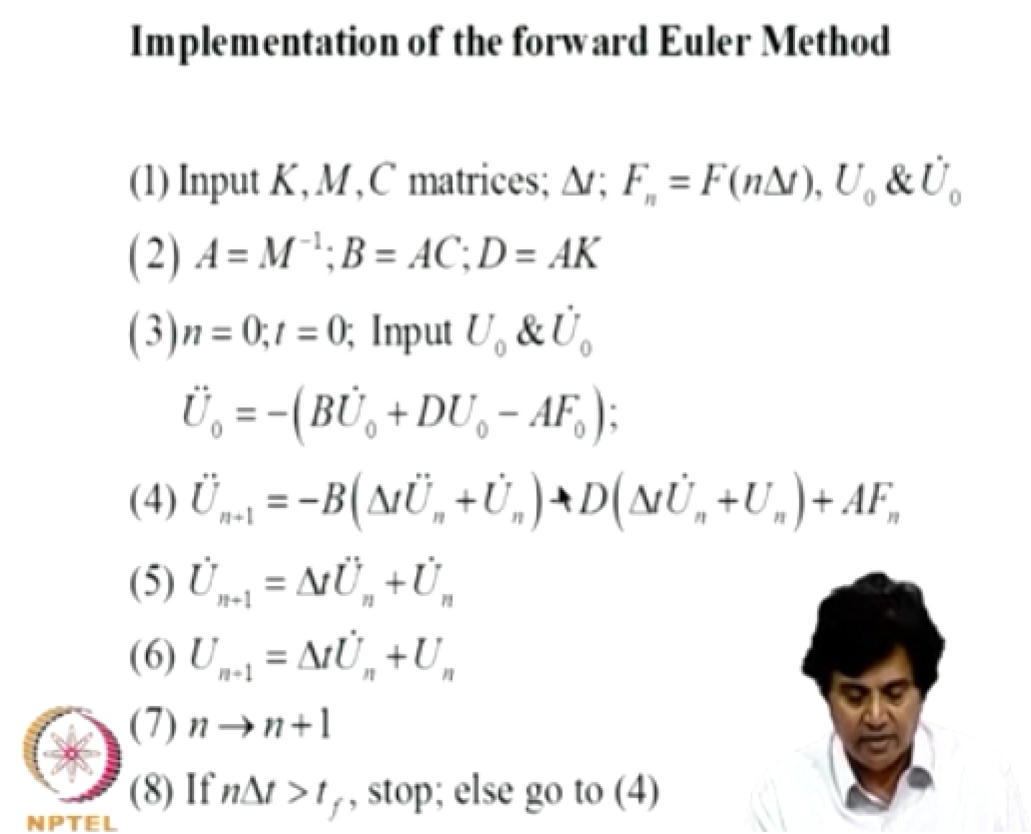
(4) $U_{n+1} = \Delta t\dot{U}_n + U_n$

(5) $n \rightarrow n+1$

(6) If $n\Delta t > t_f$, stop; else go to (2)

NPTEL

So how do we implement the method you can see here the scheme is very clear, I have to bank on this equation, this equation, and the final equation here, so I will start with T = 0, N = 0, we will input the initial condition U naught and U naught dot, I will also need the initial acceleration, I will obtain it from the equation of motion, then I start with this equation, U double dot N +1 which is given by this, so here UN double dot, UN dot etcetera are known, for example when N = 0 this will be U double dot 1, this will be U naught double dot, U naught dot and U naught, and they are already specified here, U naught and U naught dot are given initial conditions and U naught dot is obtained from the governing equilibrium equation, so this is right. Then I'll find the velocity and the displacement, then I'll increment N and if N delta T is more than TF I'll stop, otherwise I'll go to 2, so this is the simple-minded outline of how to implement the forward difference scheme, but if you carefully look at this it is not necessary to invert M matrix at every time T, you need not have to multiply M inverse and C at every time T and so on and so forth, so we can refine this a bit so what I will do is I will calculate these matrices inverse and this product outside the loop, they need to be done only once, we need not do every time T, so if you do that I get a more usable implementation scheme, so I will input



KMC matrices and delta T there is some algorithmic parameter and we will also store all the excitation and then the initial conditions, so I will define A as M inverse, B as A into C, C is damping, D is A into K.

Now we will start with N = 0, and we will accept this U naught and U naught dot form U naught double dot, so I am not inverting any of these matrices nor I am multiplying any matrix now, so UN+1 double dot is given by this, I get similarly velocity and displacement, and I will increment time and if I cross the final time instant I will stop, otherwise I will restart with this time, so the most of the calculation gets done in the steps 4 to 6, and I am not repeating any calculation, if I can avoid I'll avoided all that, okay, so this is the forward Euler forward difference scheme, okay, it is logically simple, so what we can do now is we can think of asking

Forward Euler Method : stability analysis

$$\ddot{x} + 2\eta\dot{x} + \omega^2 x = f(t) \Rightarrow$$

$$\dot{U}_{n+1} = \Delta t\ddot{U}_n + \dot{U}_n = \Delta t\left(-2\eta\omega\dot{U}_n - \omega^2 U_n + f_n\right) + \dot{U}_n$$

$$U_{n+1} = \Delta t\dot{U}_n + U_n$$

$$\Rightarrow \begin{pmatrix} U_{n+1} \\ \dot{U}_{n+1} \end{pmatrix} = \begin{bmatrix} 1 & \Delta t \\ -\omega^2\Delta t & 1-2\eta\omega\Delta t \end{bmatrix} \begin{pmatrix} U_n \\ \dot{U}_n \end{pmatrix} + \begin{pmatrix} 0 \\ \Delta t f_n \end{pmatrix}$$

$$\Rightarrow Y_{n+1} = AY_n + L_n$$

$$Y_n \to Y_n + \Gamma_n$$

$$\Rightarrow Y_{n+1} + \Gamma_{n+1} = AY_n + A\Gamma_n + L_n$$

$$\Rightarrow \Gamma_{n+1} = A\Gamma_n$$

how does errors behave, when I apply this scheme to analysis of simple systems.

Now for purpose of illustration I will consider a single degree freedom system under some excitation F(t), now this is the scheme I will get using forward difference scheme for this problem I will get this is the scheme, okay, now this can be written as capital YN+ 1 as AYN+LN okay, where A is this matrix, okay, now suppose at the N-th step the state that is UN+1 and UN+1 dot is contaminated by noise gamma N, now the question is how does gamma N behave? So I will substitute this into this equation so YN+ 1 will be YN+1 + gamma N+1 = AYN + A gamma N + LN, so now YN+1 is solution to this equation, therefore this, this and this cancel out, they are equal, the sum of these two is equal to this, so that gets out of reckoning, so the error is gamma N+1 is A gamma N, so this is the equation for evolution of the error gamma, so we will digress a bit now, will return to that shortly, but let us examine the

Digression

Consider the scalar equation $x_{n+1} = ax_n$ with $x_0 \neq 0$ sepecified.

$\Rightarrow$

$$x_1 = ax_0$$
$$x_2 = ax_1 = a^2 x_0$$
$$\vdots$$
$$x_n = ax_{n-1} = a^n x_0$$

Clearly

$$\lim x_n \left.\begin{matrix} \to 0 \text{ if } |a| < 1 \\ = x_0 \text{ if } |a| = 1 \\ \to \infty \text{ if } |a| > 1 \end{matrix}\right.$$

nature of this kind of finite difference equations suppose you consider a scalar equation XN+1 is AXN, and we start with some nonzero initial conditions, so X1 will be AX naught, X2 will be AX 1 which is A square X naught, and XN will be A to the power of N X naught, now if I am interested in knowing what happens to XN as N tends to infinity, we can easily see here that this function will go to 0, if modulus of A is less than 1, suppose A is 0.1, A square will be 0.01, A to the power of 3 will be 0.001 and so on and so forth, A to the power of 10 will be 10 to, you know 1 into 10 to the power of – 10, so on and so forth, so this is the condition.

Now if A is 1, + 1 or – 1, XN will stay at X naught, as N tends to infinity, okay, and similarly if A is greater than 1, suppose A is 2, initially it will be 2X naught, then 4X naught, then 16 X naught and so on and so forth, it goes to infinity as N tends to infinity, so the behavior of this XN as N tends to infinity is crucially governed by the value of A, if the absolute value of A is less than 1, XN goes to 0 as N tends to infinity, this is scalar equation. Suppose now you

Let us now consider the $s \times 1$ vector equation $x_{n+1} = Ax_n$

with $x_0 \neq 0$ sepecified. Here A is a $s \times s$ matix.

$$\Rightarrow$$

$$x_1 = Ax_0$$

$$x_2 = Ax_1 = A^2 x_0$$

$$\vdots$$

$$x_n = Ax_{n-1} = A^n x_0$$

Let us introduce the transformation $x_n = \Phi z_n$

$$\Rightarrow \Phi z_{n+1} = A\Phi z_n$$

Let Φ be such that $\Phi'\Phi = I \;\&\; \Phi' A\Phi = \text{Diag}\left[\lambda_i\right]$

We can select Φ by solving the eigen value problem associated with matrix A.

37

consider S cross 1 vector equation, suppose this is XN + 1 = AXN, where A is S cross S matrix, okay. Now X1 is AX naught, X2 is AX1 which is A square X naught, similarly XN will be A to the power of N, X naught. Now let's do the following, let us introduce a transformation XN = phi ZN, so this equation for XN+1 will be phi ZN +1 will be equal to A phi ZN.

Now let phi be such that phi transpose phi is identity, and phi transpose A phi is a diagonal matrix, say if A is such that this is possible, how do we select, we can select phi by solving the eigenvalue problem associated with A, that we have seen in few vacations how to do that, now suppose A, B such that this is possible, then I can write phi ZN + 1 = A phi ZN and pre

$$x_n = \Phi z_n$$
$$\Rightarrow \Phi z_{n+1} = A\Phi z_n$$
$$\Rightarrow \Phi'\Phi z_{n+1} = \Phi' A\Phi z_n \Rightarrow z_{n+1} = \mathrm{Diag}\left[\lambda_i\right] z_n$$
$$\Rightarrow z_{n+1}^k = \lambda_k z_n^k; k = 1, 2, \cdots, s$$
$$\Rightarrow z_{n+1}^k = \lambda_k^n z_0^k; k = 1, 2, \cdots, s$$
$$\Rightarrow x_n^j = \sum_{k=1}^{s} \Phi_{jk} \lambda_k^n z_0^k \text{ with } z_0 = \Phi' x_0$$

We are interested in whether or not $\lim_{n\to\infty} \left| x_n^j \right| \to 0$ for $j = 1, 2, \cdots s$

Clearly,

$$\max_{1\le k\le s} \left|\lambda_k\right| < 1 \Rightarrow \lim_{n\to\infty} \left| x_n^j \right| \to 0 \text{ for } j = 1, 2, \cdots s$$

That is, the behavior of $\lim_{n\to\infty} \left| x_n^j \right|$ for $j = 1, 2, \cdots s$ controlled by the highest modulus of the eigenvalues of A.

multiply by phi transpose I get this equation, now since phi transpose phi is I, I get ZN+1 and phi transpose A phi is a diagonal matrix of lambda I, I get this, okay, so Z is a S cross 1 vector therefore I can write the K-th element of that, ZN +1K as lambda K ZNK where K runs from 1 to S, so this is the equation I get, ZN+1 K, now the behavior of this individual Z as K tends to infinity, depends on absolute value of lambda K, the K-th eigenvalue, right, now if you are interested in J-th component of your vector X, that is given by this summation, okay, so the behavior of this as N tends to infinity crucially depends on the highest eigenvalue that if you rank order this lambda K depending on the absolute value of its, that is the absolute value of lambda K, we are interested in, if you are interested in knowing what happens to this modulus of XNJ as N tends to infinity, for this to go to 0 we require that the maximum value of this lambda K must less than 1, okay, because each 1 is a scalar equation, for each scalar equation to go to 0, the each of the eigenvalue, absolute value should be less than 1, so if the highest eigenvalue is a modulus is less than 1, then XNJ will go to 0, as N tends to infinity, so therefore if you are interested in behavior of XNJ as N tends to infinity, that is controlled by the highest modulus of the eigenvalues of A, okay.

Alternatively, consider

$x_n = Ax_{n-1} = A^n x_0$

Let us seek solution in the form $x_n = \alpha^n x_0$

Here α is scalar which is in general complex valued.

$x_n = \alpha^n x_0 \Rightarrow x_{n+1} = \alpha^{n+1} x_0$

Consider the equation $x_{n+1} = Ax_n = A^n x_0$

$\alpha^{n+1} x_0 = A\alpha^n x_0 \Rightarrow Ax_0 = \alpha x_0$

For nontrivial solutions $|A - \alpha I| = 0$

$\Rightarrow \alpha_i = \alpha_{0i} \exp(i\theta_i)$

Condtion for $\lim_{n\to\infty} |x_n^j| \to 0$ for $j = 1, 2, \cdots s$ is given by $\max_{1\le i\le s} |\alpha_i| < 1$

By definition $\max_{1\le i\le s} |\alpha_i| = \rho(A)$ is called the spectral radius of A.

NPTEL

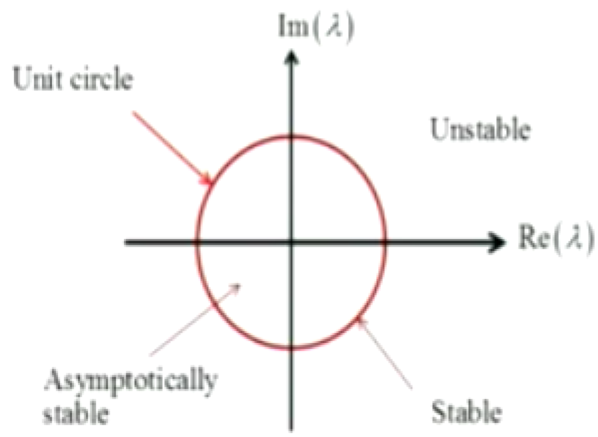
39

Now alternatively we can also consider XN as AXN + 1 into A into XN-1, I can write it as A to the power of N, X naught, now if we seek the solution now of the form XN is alpha N into X naught, suppose if you seek this solution, for some value of alpha the such a solution may be possible, for which value of alpha this is possible we have to see, this alpha could be scalar, the scalar which can, in general be complex value, so now you substitute this into this equation, I get XN is alpha N X naught, therefore XN+1 is alpha N+1 X naught. Now if you consider this equation XN + 1 is AXN, A to the power of X naught and make these substitutions I get alpha N + 1 X naught is A alpha N X naught, so from this I get AX naught is alpha X naught, so this would mean for non-trivial solutions the determinant of A - alpha I must be equal to 0, and if we represent the roots of this equation which are the eigenvalues of A, suppose I-th route is written as alpha 0I exponential I theta I, this is a complex number so I can always represent like this.

The condition for XNJ modulus of that as N tends to infinity to go to 0 is given by the condition that the maximum value of, absolute value of alpha I must be less than 1, so that would mean actually this quantity which is a maximum value of modulus of the eigenvalue for I running from 1 to S is known as spectral radius of A, okay so what we are asking is the spectral radius

$$\max_{1\le k\le s}|\lambda_k| < 1 \Rightarrow \lim_{n\to\infty}|x_n^j| \to 0 \text{ for } j = 1,2,\cdots s$$

$\Rightarrow$ The roots of the characteristic equation must lie within the unit circle in the complex plane.



We do not need the value of the eigenvalues.

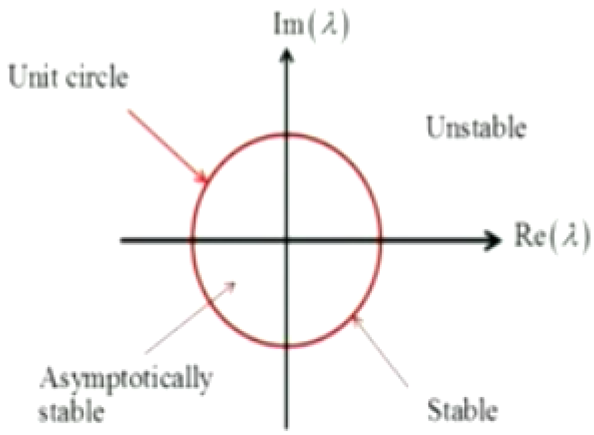
We need to only ascertain if all the roots of the characteristic equation lie within the unit circle in the complex plane.

of A must be less than 1 so that would mean if you look at the eigenvalue in the complex plane we have real part here, imaginary part here, and this is the so-called unit circle, okay it has radius equal to 1, what we want is the root should be inside this unit circle for stability asymptotic stability, if it is right on the unit circle the errors don't grow, but it is stable, but if it is outside the unit circle the errors would grow, okay, so this is what we need to verify if you are interested in studying the growth of errors.

One more thing that we should notice here is the only question we need to answer is whether roots lie within unit circle or not, we are not so much interested in knowing the absolute value of the route, we are simply interested in knowing a qualitative feature associated with the roots. So we do not need the value of eigenvalues, we need to only ascertain if all the roots of the characteristic equation lie within the unit circle in the complex plane, given that we are asking a

$$\max_{1\le k\le s}|\lambda_k|<1 \Rightarrow \lim_{n\to\infty}|x_n^j| \to 0 \text{ for } j=1,2,\cdots s$$

$\Rightarrow$ The roots of the characteristic equation must lie within the unit circle in the complex plane.



We do not need the value of the eigenvalues.

We need to only ascertain if all the roots of the characteristic equation lie within the unit circle in the complex plane.

40

simple question then actually not asking, we are not asking, we are not interested in determining the value of all the eigenvalues, we are simply interested in knowing whether all eigenvalues lie within the unit circle or not, to answer this simpler question we can develop an alternative method which is computationally simpler than finding all the roots and finding out whether they lie within unit circle or not, so that takes us to discussion on what is known as Jury's criterion where given a polynomial by using Jury's criterion we can verify whether roots of a polynomial lie within a unit circle or not, so that has bearing on discussion on stability of the finite difference time integration schemes, so what we will do is, we will take up that issue in the next lecture, and we will conclude this lecture at this stage.

Programme Assistance

Guruprakash P
Dipali K Salokhe

Technical Supervision
B K A N Singh
Gururaj Kadloor

Indian Institute of Science

Bangalore