**Indian Institute of Science**
**Bangalore**

**NP-TEL**
**National Programme on**
**Technology Enhanced Learning**

**Copyright**

**1. All rights reserved. No part of this work may be reproduced, stored or transmitted in any form or by any means, electronic or mechanical, including downloading, recording, photocopying or by using any information storage and retrieval system without prior permission in writing from the copyright owner.**

**Provided that the above condition of obtaining prior permission from the copyright owner for reproduction, storage or transmission of this work in any form or by any means, shall not apply for placing this information in the concerned institute's library, departments, hostels or any other place suitable for academic purposes in any electronic form purely on non-commercial basis.**

**2. Any commercial use of this content in any form is forbidden.**

**Course Title**

# Finite element method for structural dynamic And stability analyses

## Lecture – 16
## Energy conservation Nonlinear systems.

**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

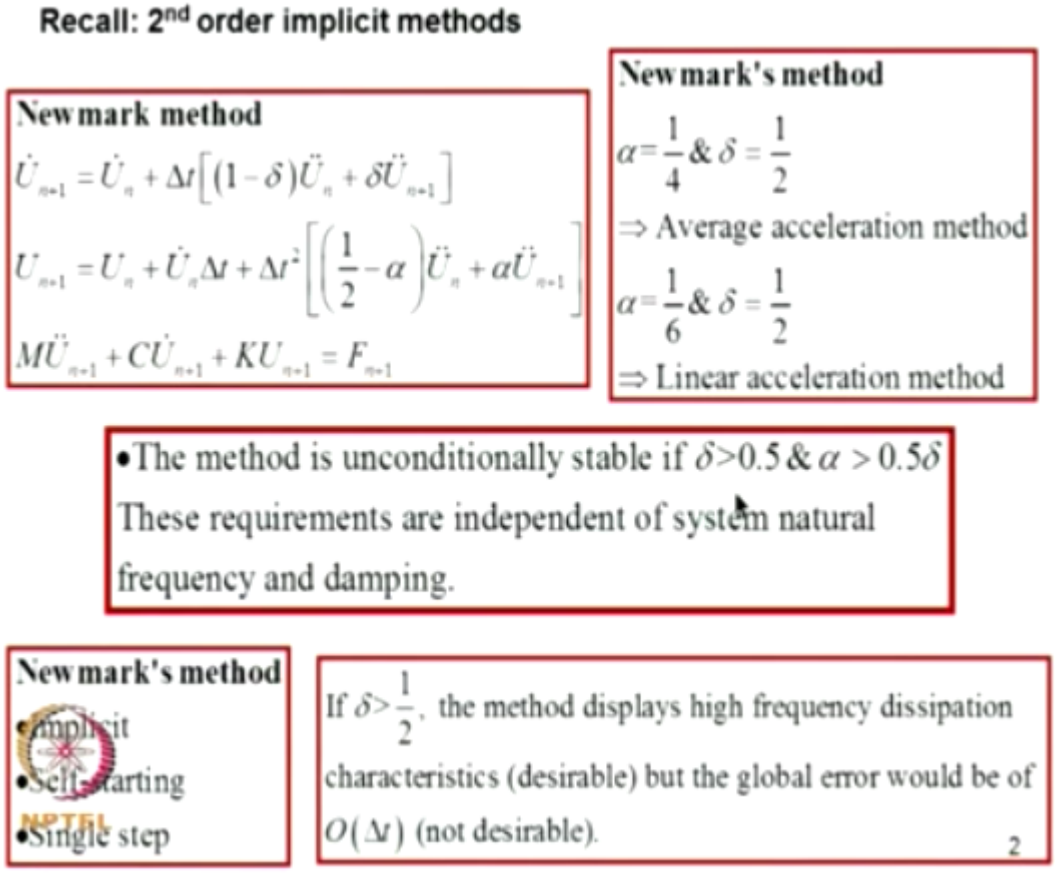
Lecture-16 Energy conservation. Nonlinear systems.

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

NPTEL

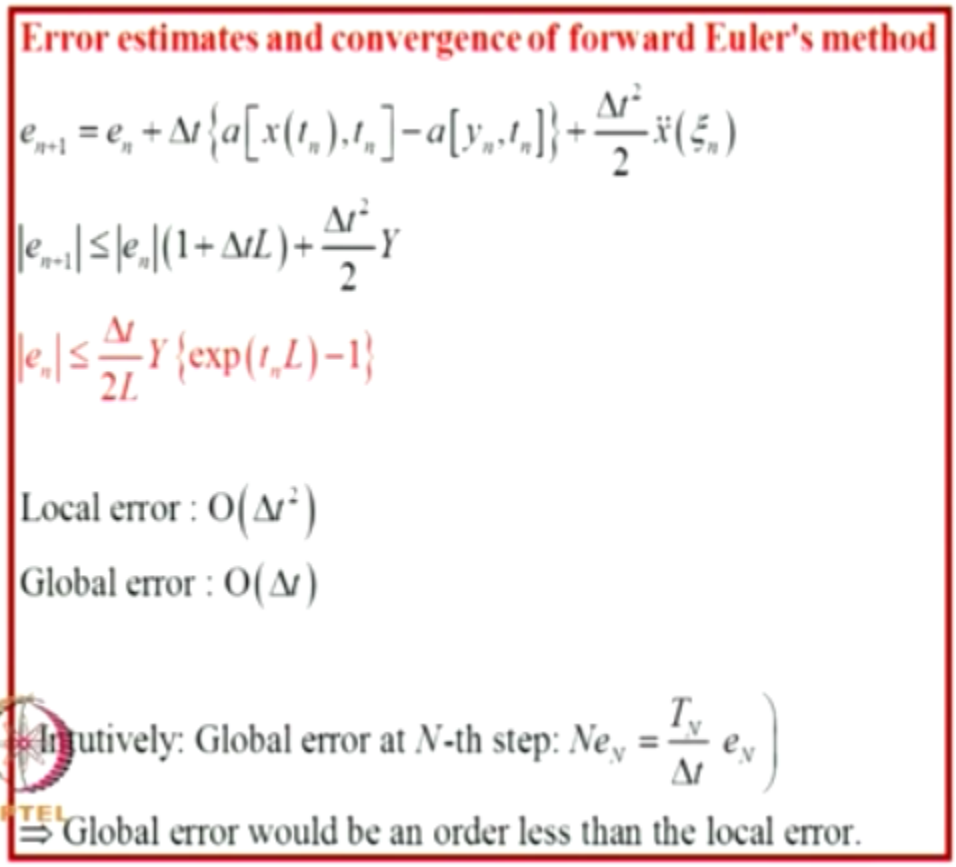


We'll continue with our discussion on methods for integrating equations of equilibrium in dynamics. In today's lecture we will talk about the concept of energy conservation and also address few issues related to nonlinear systems, so what we have been discussing we have




developed, discussed this Newmark's method which is a second-order implicit method, the three stage formulary for this is displayed here, we have shown that the method is conditionally stable if we select this parameter delta and alpha to satisfy this requirement, and these requirements are independent of system natural frequency and damping. So Newmark's method is implicit, self-starting, and it is a single step method.

Now we can also obtain the other methods like average acceleration method, linear acceleration method, as special cases of Newmark's method by assigning the parameters alpha and delta values as shown here, this also we have seen. Now for delta greater than half the method displays high frequency dissipation characteristic which are desirable, but the global error would be of order delta T which is not desirable.




We have also seen questions about error estimates and convergence of forward Euler method, we have shown that the error is of the, is bounded by this quantity, so we saw that local, although the local error is order delta T square the global error is order delta T, this is a common feature in these algorithms intuitively the explanation for this can be this, global error at nth step, if we write it as, if EN is a local error the global error at the end of nth step will be N into EN, so N can be written as TN/delta T, so if EN is of order delta T square as it is here when divided by delta T it becomes order delta T, that means you have to sum up to all the errors up to that time instant that reduces the order of accuracy, so global error typically would be an order less than the local error.

**HHT $\alpha$ method and generalizations**

$$(1-\alpha_m)M\ddot{U}_{n+1}+\alpha_m M\ddot{U}_n+(1-\alpha_f)C\dot{U}_{n+1}+\alpha_f C\dot{U}_n+(1-\alpha_f)KU_{n+1}+\alpha_f KU_n$$
$$=(1-\alpha_f)F_{n+1}+\alpha_f F_n$$

$$\dot{U}_{n+1}=\dot{U}_n+\Delta t\left[(1-\delta)\ddot{U}_n+\delta\ddot{U}_{n+1}\right]$$

$$U_{n+1}=U_n+\dot{U}_n\Delta t+\Delta t^2\left[(0.5-\alpha)\ddot{U}_n+\alpha\ddot{U}_{n+1}\right]$$

Parameter range for unconditional stable scheme

$$\alpha_m\le\alpha_f\le 0.5;\delta=0.5-\alpha_m+\alpha_f;\alpha\ge 0.25+0.5(\alpha_f-\alpha_m)$$

The method possesses desirable high frequency dissipation characteristics, is unconditionally stable, and second order accurate.




The HHT alpha method and generalizations overcome one weakness of Newmark's method that to be, to possess high frequency numerical dissipation the method would become only order delta T accurate, so what this HHT alpha method does is, it introduces additional algorithmic parameters, for example in a generalized version we have alpha M, alpha F, which are additional parameters, in addition to delta and alpha that are used in Newmark's method, so we have now 4 parameters, so we have greater flexibility to you know design the algorithm to display desired properties, so it can be, we have seen that the method possesses desirable high frequency dissipation characteristics it is unconditionally stable and it has second-order accuracy.

**Desirable features of numerical integration schemes**

- At least second order accuracy
- Unconditional stability when applied to LTI systems
- Controllable algorithmic damping in higher modes

(by changing parameters other than step size)

No distortion of the contributing modes.

Suppresion of spurious modes.

Investigate spectral radius as frequency $\to \infty$

(Does it go to zero?)

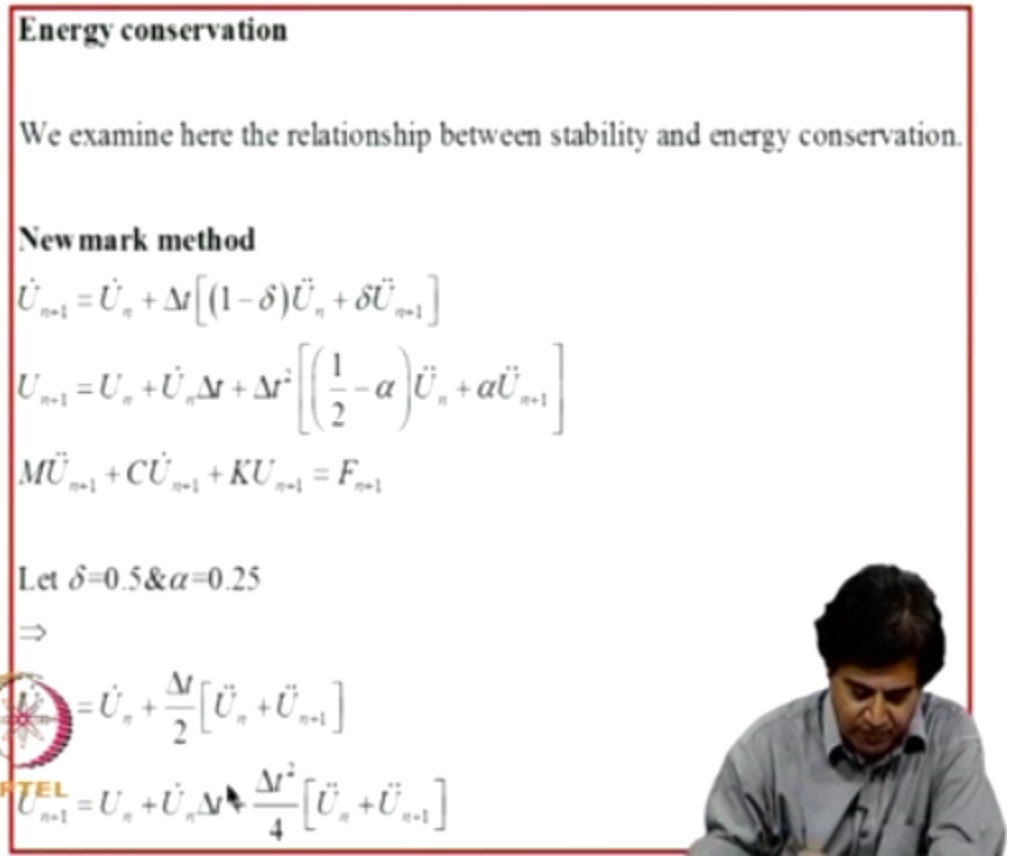
Spectral radius $\to 1$ as driving frequency $\to 0$

- No overshoot
- Excessive oscillations during the first few steps
- Self-starting
- No more than one set of implicit equations to be solved at each step

So we can summarize what are the desirable features of a numerical integration scheme, so it should at least have second-order accuracy and it should be unconditionally stable when applied to linear time-invariant systems, and we want to have controllable algorithmic damping in higher modes that means by changing parameter other than the step size, that means these parameters are algorithmic parameter not the system parameters we should be able to ensure that the numerical damping that will be present in the system you know does not distort the lower order contributing modes, and it suppresses the spurious higher-order modes, so again this requires investigation into spectral radius of the amplification matrix as frequency tends to infinity, for large frequency the spectral radius should go to zero so that higher order, high-frequency components are dissipated and the spectral radius should be as close to 1 for low frequency region where there will be participating modes which are contributing significantly to the response, and also there should be no overshoot that means excessive oscillations during the first few steps should not be present ideally the scheme should be self-starting and when we deal with nonlinear systems as we will do today later, no more than one set of implicit equations need to be solved at each step, so the concerns here are let me summarize if you are carrying out a dynamic analysis of a structure you make a finite element model while selecting the size of the model you have to keep in mind the highest frequency that is participating, highest frequency that is present in the excitation then suppose that highest frequency is say 20 hertz as perhaps in the case of an earthquake engineering problem then we should ensure that the finite element model that we make should have acceptable accuracies on Eigen characteristics maybe up to say 50 hertz or 80 hertz 3 to 4 times the highest frequency present in the excitation, over that frequency range we should capture all the modes that are present in

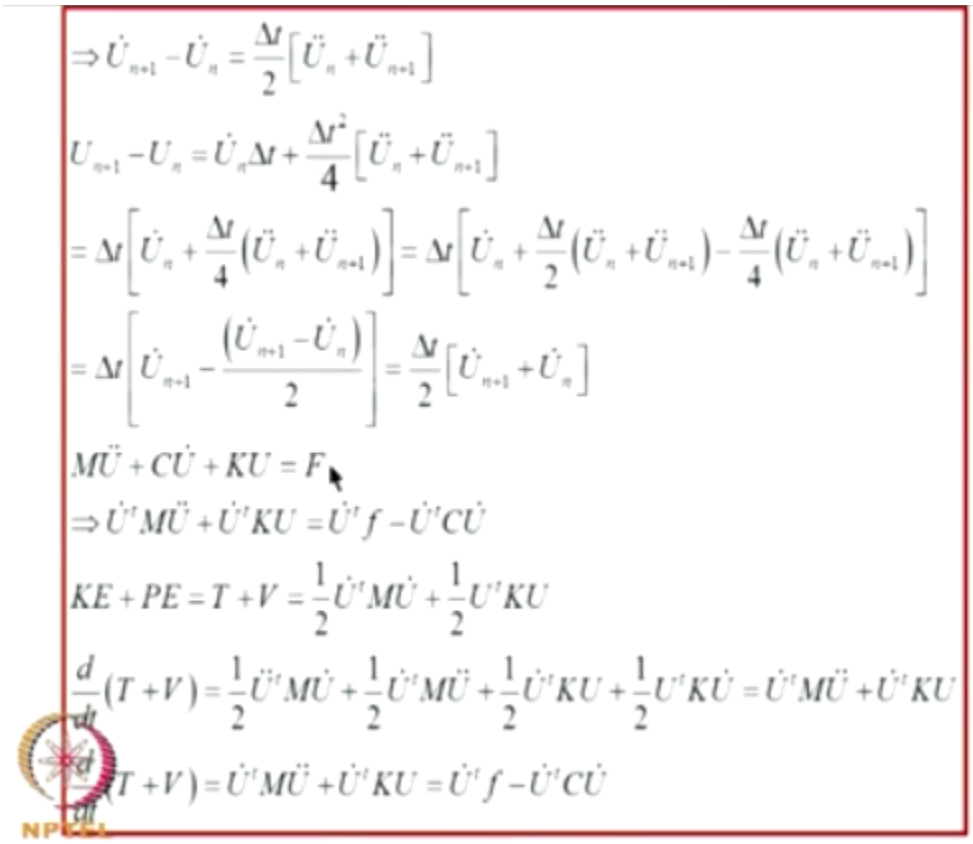
the structure with good accuracy, suppose between say 0 to 50 hertz there are say 12 modes in a structure which contributes, whose natural frequency lie in 0 to 50 hertz say for example, then our model should have at least, the degrees of freedom in our model should be at least 10 times the number of modes that you would like to retain in the model expansion.

So in order that we get acceptable accuracy on the lower order model characteristics, our model size need to be large, consequently there will be higher order spurious modes which are numerically not accurate and in a physical sense they don't really contribute to the response, so the algorithm that we use should ensure that those spurious modes are dissipated through numerical dissipation, and the contributing lower order modes are not distorted, so that is the main issue when it comes to discussion on algorithmic dissipation.




Now, we will now consider what is the relationship between stability and energy conservation? We have talked about the so-called spectral stability, you know the integration schemes unapplied to linear time-invariant systems, let's ask the question now, those concerns are related to conservation of energy.

So let's focus our discussion on Newmark's method, so in the Newmark method we have this set of 3 formulae, this is for velocity, this is for displacement, and this is the equilibrium equation at time TN+1, now for purpose of discussion let us take delta to be 0.5, and alpha to be 0.25, we can do a general discussion but let us simplify our study a bit, we will use this representation, so in which case UN, velocity is given by this, and displacement would be given by this. Now we can reorganize some of these terms, suppose I write UN+ 1 dot – UN dot, this




7

is given by this. Similarly UN + 1 - UN is given by this. Now, this we can simplify and we can show that UN+1 - UN is given by this.

Now we have the equilibrium equation at time T, now what I will do is I will multiply by U dot transpose, pre multiply by U dot transpose so this equation is what we get. Now if U dot transpose into F(t) is actually the work done by external force per unit time, for example in a single degree freedom system velocity into force will be work done per unit time, force into displacement is a work done, therefore force into velocity will be work done per unit time which is the power input, so this is basically the power balance equation where we are equating the power input to the system to the contributions from inertial forces, stiffness and damping forces. Now that sum of kinetic energy and potential energy will now consider that it will be T+V, and T is 1/2 U dot transpose MU dot, V is 1/2 U transpose KU this we have seen. Now the time variation of the total energy that D/DT (T+V) if you are interested we can differentiate this, so we will get U double dot transpose MU dot + 1/2 U dot transpose MU double dot, similarly the other two terms.

Now this is a scalar quantity U dot transpose KU is a scalar quantity, so is U transpose KU dot, U is N cross 1, K is N cross N, this is N cross 1,. so U transpose will be 1 cross N, so this will be a scalar quantity, so for this we can as well write U dot transpose KU, you can't replace it by this transpose, so we can or in other words these two can be added and these two can be added and we can write in this form. So the time variation of total energy is therefore given by this.

$$\frac{d}{dt}(T+V) = \dot{U}^t M\ddot{U} + \dot{U}^t KU = \dot{U}^t f - \dot{U}^t C\dot{U}$$

$$\Rightarrow (T+V)_{t_n}^{t_{n+1}} = \int_{t_n}^{t_{n+1}} \left(\dot{U}^t f - \dot{U}^t C\dot{U}\right) dt$$

$$T_{n+1} = \frac{1}{2}\dot{U}_{n+1}^t M\dot{U}_{n+1} \;\&\; T_n = \frac{1}{2}\dot{U}_n^t M\dot{U}_n$$

$$T_{n+1} - T_n = \frac{1}{2}\left(\dot{U}_{n+1}^t M\dot{U}_{n+1} - \dot{U}_n^t M\dot{U}_n\right)$$

$$= \frac{1}{2}\left(\dot{U}_{n+1}^t M\dot{U}_{n+1} + \dot{U}_{n+1}^t M\dot{U}_n - \dot{U}_{n+1}^t M\dot{U}_n - \dot{U}_n^t M\dot{U}_n\right)$$

$$= \frac{1}{2}\left(\dot{U}_{n+1}^t M\dot{U}_{n+1} + \dot{U}_{n+1}^t M\dot{U}_n - \dot{U}_n^t M\dot{U}_{n+1} - \dot{U}_n^t M\dot{U}_n\right)$$

$$= \frac{1}{2}\left\{\dot{U}_{n+1}^t M\left(\dot{U}_{n+1} + \dot{U}_n\right) - \dot{U}_n^t M\left(\dot{U}_{n+1} + \dot{U}_n\right)\right\}$$

$$= \frac{1}{2}\left\{\left(\dot{U}_{n+1}^t - \dot{U}_n^t\right) M\left(\dot{U}_{n+1} + \dot{U}_n\right)\right\}$$

Similarly, we get $V_{n+1} - V_n = \frac{1}{2}\left\{\left(U_{n+1}^t - U_n^t\right) K\left(U_{n+1} + U_n\right)\right\}$



Now we will now see if I now integrate this you know T+V from TN to TN+1 we will get this integral. Now at N+1 the kinetic energy will be given by this, and at TN I have this expression with you replace N+1/N, I get this. So if I now consider the increment in kinetic energy this is the increment in kinetic, so now I will simplify this, I am adding and subtracting some terms and rearranging these terms to show that this incrementing kinetic energy can be given by this quantity.

$$T_{n+1} - T_n = \frac{1}{2}\left\{\left(\dot{U}^t_{n+1} - \dot{U}^t_n\right) M\left(\dot{U}_{n+1} + \dot{U}_n\right)\right\}$$

$$V_{n+1} - V_n = \frac{1}{2}\left\{\left(U^t_{n+1} - U^t_n\right) K\left(U_{n+1} + U_n\right)\right\}$$

$$\dot{U}_{n+1} - \dot{U}_n = \frac{\Delta t}{2}\left[\ddot{U}_n + \ddot{U}_{n+1}\right]$$

$$U_{n+1} - U_n = \frac{\Delta t}{2}\left[\dot{U}_{n+1} + \dot{U}_n\right]$$

$$\Rightarrow$$

$$T_{n+1} - T_n = \frac{\Delta t}{4}\left\{\left(\ddot{U}_n + \ddot{U}_{n+1}\right)^t M\left(\dot{U}_{n+1} + \dot{U}_n\right)\right\} = \frac{\Delta t}{4}\left\{\left(\dot{U}_{n+1} + \dot{U}_n\right)^t M\left(\ddot{U}_n + \ddot{U}_{n+1}\right)\right\}$$

$$V_{n+1} - V_n = \frac{\Delta t}{4}\left\{\left(\dot{U}_{n+1} + \dot{U}_n\right)^t K\left(U_{n+1} + U_n\right)\right\}$$

$$\Rightarrow (T+V)_{t_n}^{t_{n+1}} = \frac{\Delta t}{4}\left(\dot{U}_{n+1} + \dot{U}_n\right)^t \left\{M\left(\ddot{U}_n + \ddot{U}_{n+1}\right) + K\left(U_{n+1} + U_n\right)\right\}$$

Recall $M\ddot{U} + C\dot{U} + KU = F \Rightarrow$

$$(T+V)_{t_n}^{t_{n+1}} = \frac{\Delta t}{4}\left(\dot{U}_{n+1} + \dot{U}_n\right)^t \left\{F_{n+1} + F_n - C\dot{U}_{n+1} - C\dot{U}_n\right\}$$

Now similarly we get VN+1 – VN as this quantity, so we have TN+1 -TN as this, VN+1 - VN as this, and increments in velocity and displacements are given by this, so these terms are here now this is according to Newmark's method, so now I will substitute these terms into the expression for increment in kinetic energy and potential energy, there is a Newmark beta, Newmark's approximation, so upon doing that I get this as the increment in kinetic energy, this is the increment in strain energy, so consequently the D/DT of total energy is given by this.

Now therefore if we now recall MU double dot + U dot + K = F from this we can write this expression, okay that means I am looking for T+V at TN+1 and TN, I get this, so this is the

$$(T+V)_{t_n}^{t_{n+1}} = \frac{\Delta t}{4}\left(\dot{U}_{n+1}+\dot{U}_n\right)^T\left\{F_{n+1}+F_n-C\dot{U}_{n+1}-C\dot{U}_n\right\}$$

We have $U_{n+1}-U_n = \frac{\Delta t}{2}\left[\dot{U}_{n+1}+\dot{U}_n\right]$

$$\Rightarrow (T+V)_{t_n}^{t_{n+1}} = \frac{1}{2}\left(U_{n+1}-U_n\right)^T\left(F_{n+1}+F_n\right)-\frac{\Delta t}{4}\left(\dot{U}_{n+1}+\dot{U}_n\right)^T C\left(\dot{U}_{n+1}+\dot{U}_n\right)$$

**Remark**

- For free vibration of undamped systems $(C=0 \,\&\, F=0)$ we get $(T+V)_{t_n}^{t_{n+1}} = 0$.

That is, the Newmark algorithm conserves the total energy.

This is true for any step size.

- Recall that the method is unconditionally stable if $\delta > 0.5 \,\&\, \alpha > 0.5\delta$

These requirements are independent of system natural frequency and damping.

It can be shown that for this case that the energy balance over one time step is negative.

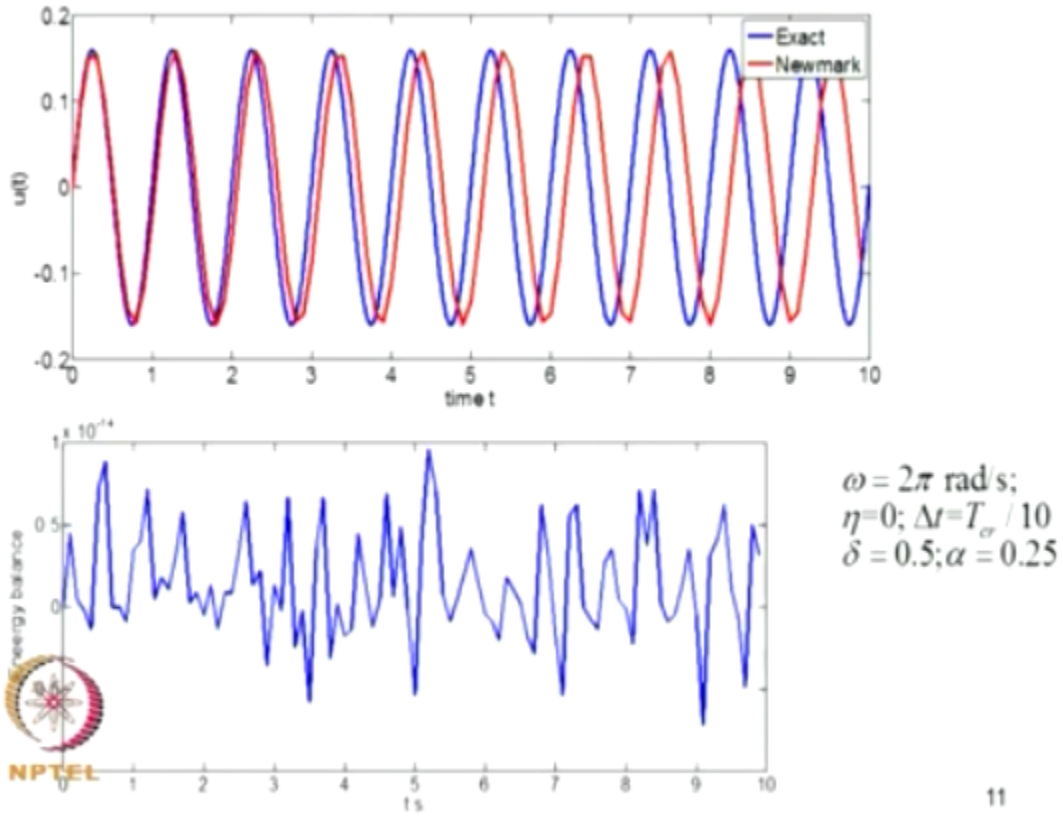
- In regions in the $\alpha - \delta$ space where the method is not stable, the energy balance over a time step turns out to be positive.

10

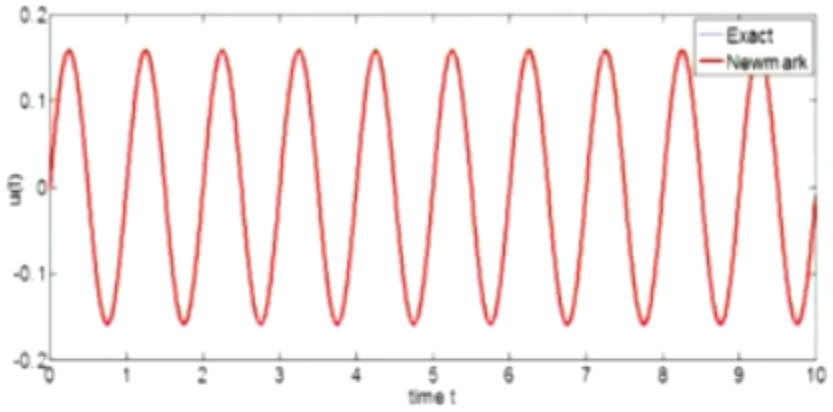
expression we have got, and now this is again Newmark beta approximation for displacement if I now substitute that I get the change in total energy from time step TN to TN+1 is given by this.

Now if we talk about free vibration F will be 0, and C will be 0, undamped free vibration, then I see that the change in total energy is 0, which is what we expect from a conservative system, so for free vibration of undamped system that is C = 0 and F = 0 we get T + V, the change in T+V from TN to TN+1 is 0, that is the Newmark algorithm conserves the total energy, this is true for any time step size.

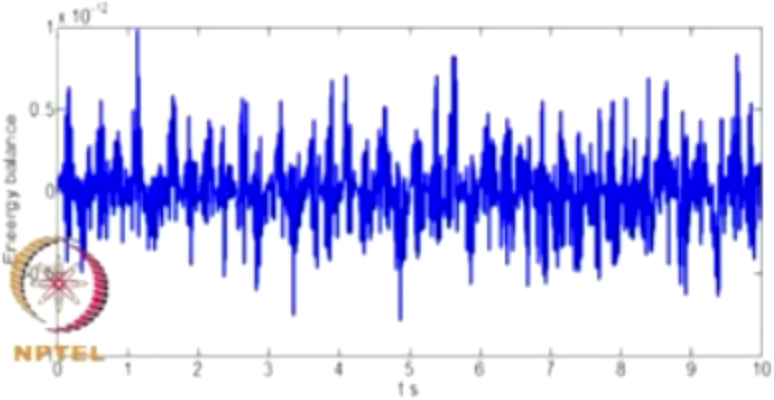
Now we have seen that the method is unconditionally stable if this condition is satisfied, these requirements are independent of system natural frequency and damping that also we have seen, we can show that for this case the energy balance or one time steps is negative, okay in general when there is a damping as well as forcing, in regions in the alpha delta space where the method is not stable the energy balance or a time step turns out to be positive, please bear in mind that the formulary I have derived is for specific choice of delta and alpha, so you could re-derive the whole story by keeping delta and alpha as it is and you can see how the question about change in energy can be tackled.




So some quick illustrations a single degree freedom system with natural frequency 2 pi, this is damping is 0 free vibration problems were considering, and I am considering step size of one tenth of critical step size and this is the algorithmic parameters, so blue line here is the exact solution, and the red one is a Newmark solution, obviously it's not accurate but it is stable. So if I now plot the energy balance we see that this is 10 to the power of -14 so you can see that the energy balance criteria is satisfied.

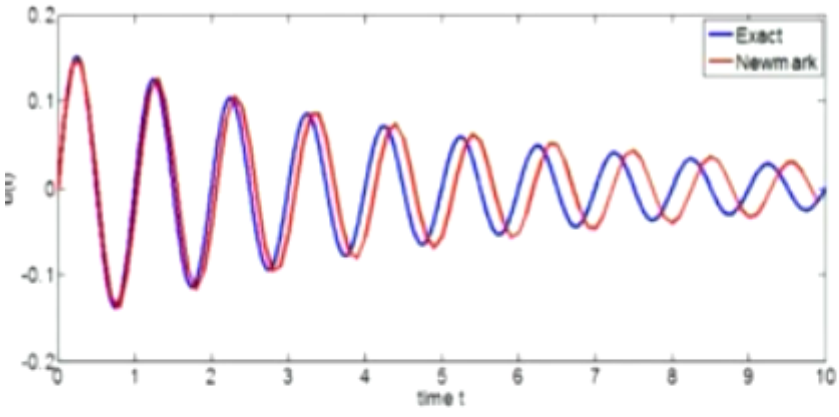



$\omega = 2\pi$ rad/s;
$\eta$=0; $\Delta t = T_{cr}$ / 100
$\delta = 0.5; \alpha = 0.25$

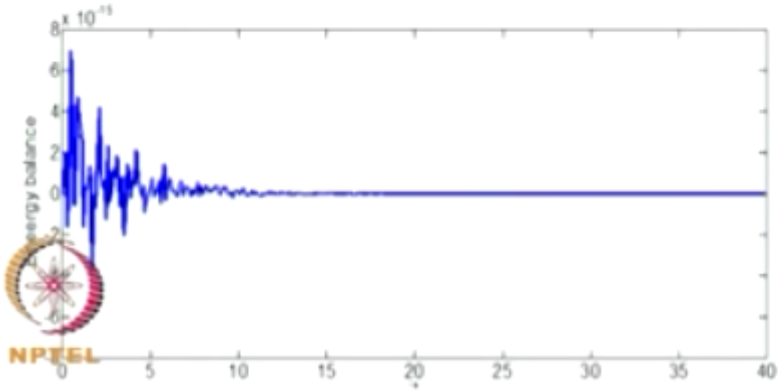



12

Similarly if I now take same data with higher step size now T critical is delta T is T critical/100 so we see that the exact and the Newmark solution match very well and the error is again quite small, the energy balance. Now how about damped free vibration? So I introduce 3% damping

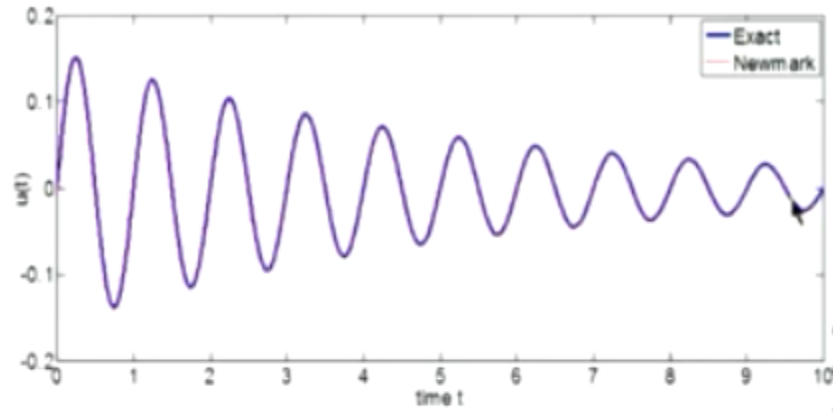



$\omega = 2\pi$ rad/s;
$\eta$=0.03;
$\Delta t = T_\sigma / 10$
$\delta = 0.5; \alpha = 0.25$

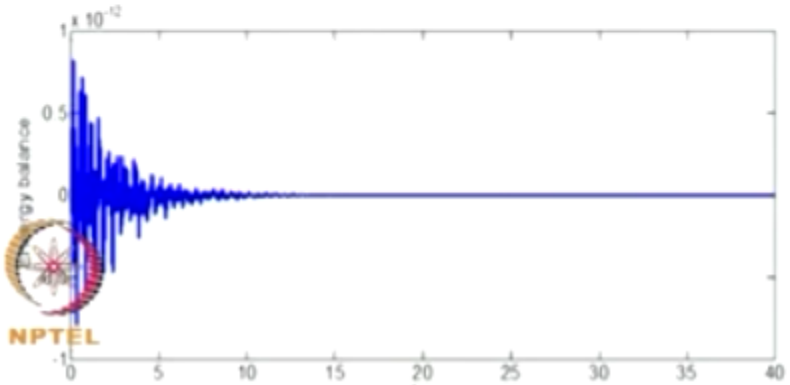



13

and again start with T critical/10 although the solution is not accurate, the error is still that the energy balance is satisfied, so energy balance is satisfied it does not mean that you have got accurate answers, okay it's like telling if you are getting stable solution you need not ensure that the solution has acceptable accuracy. Now if I increase step size to T critical/100 I get you

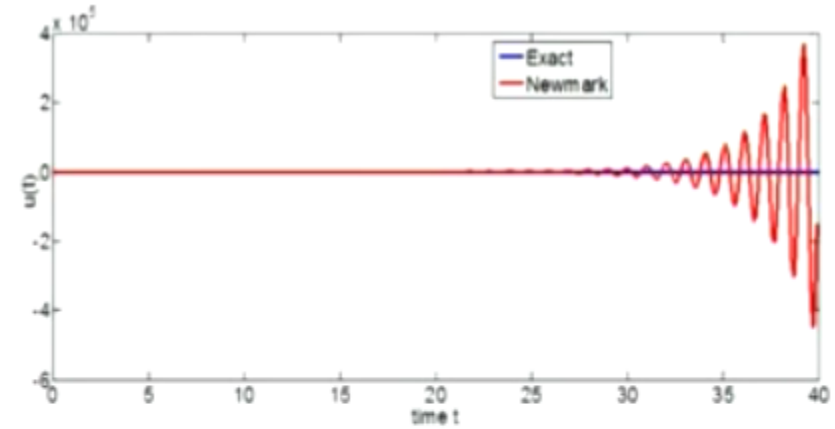



$\omega = 2\pi$ rad/s;
$\eta$=0.03;
$\Delta t = T_\sigma / 100$
$\delta = 0.5; \alpha = 0.25$

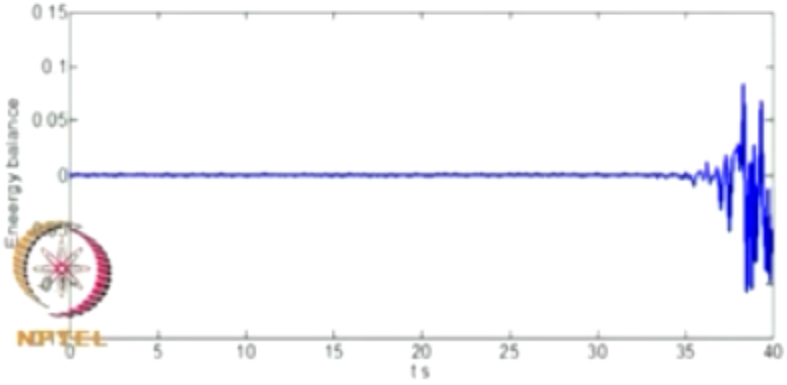





know the two exact solution and Newmark solution math, and this is the behavior of the error.

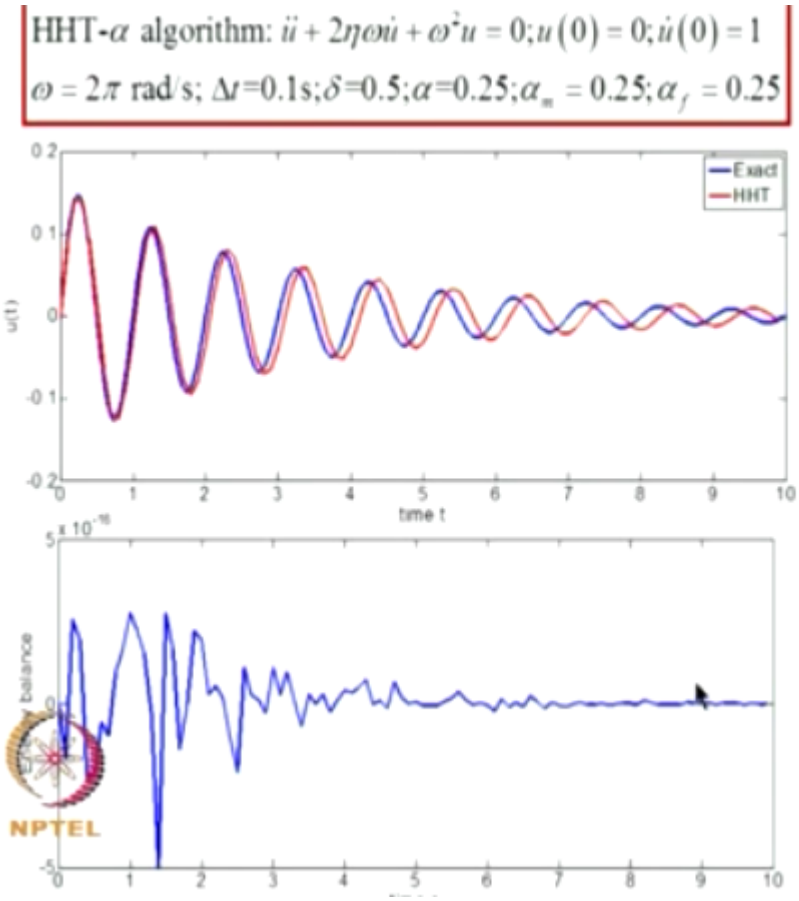



$\omega = 2\pi$ rad/s;
$\eta$=0.03;
$\Delta t = T_\sigma / 100$
$\delta = 0.2; \alpha = 0.08$



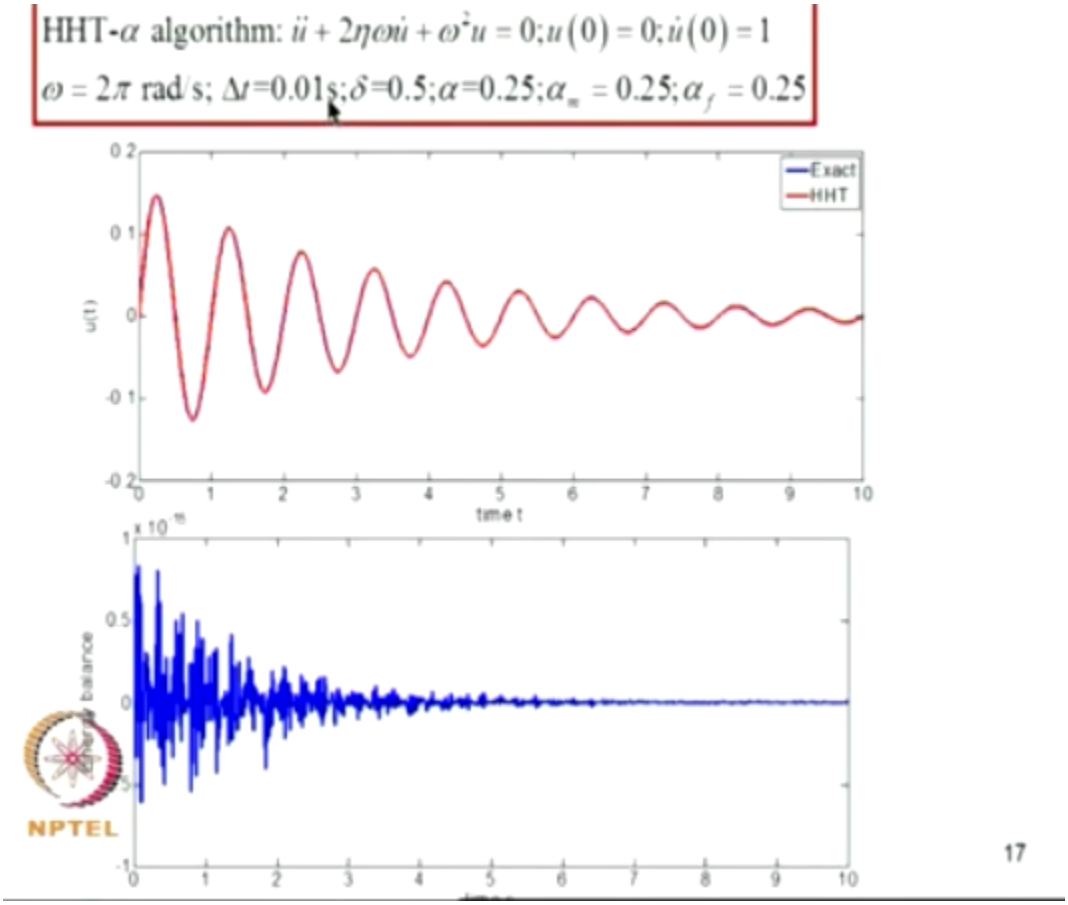



Now I deliberately select the algorithmic parameter in regions where the method is not stable, so we see that the Newmark solution diverges as time increases, and you can see that the energy balance condition is also violated, now this is, numbers are very large is 0.15, if you look at the previous graph these numbers were 10 to the power of - 12 this is very large number and it is also blowing up.






The same issue about algorithmic energy conservation can also be studied for other integration scheme, so I have shown some illustrations here this is the HHT alpha method with these parameters a damped free vibration with step size of 0.1 second and we see that the energy balance condition is met and it so happens that the step size is in the ensure stability of the solution.




Now with improved step size here for this step size there are 10 points in a period because omega is 2 pi that means period is 1 second, I have taken 10 points in a cycle that does not lead to acceptable solutions.

On the other hand if you take 100 points the two, the exact solution and HHT alpha solution matches and the error is acceptable.

S Krenk, 2006, Energy conservation in Newmark based time integration algorithms, Computer methods in applied mechanics and engineering, 195, 6110-6124

18

Now the discussion on energy conservation in Newmark based time integration algorithms has been presented in this work by Krenk, I have given the reference so in this paper he has derived that discussed this issue for both Newmark's method HHT alpha and generalized alpha methods, okay.

**Nonlinear systems : preliminary ideas**

Sources of nonlinearity

- Nonlinear strain displacement relations
- Nonlinear stress strain relations
- Nonlinear energy dissipation mechanisms

For example

Friction at joints

Aerodynamic damping

Let us consider, for the purpose of illustration, equation of motion of the form

$$M\ddot{U} + C\dot{U} + KU + g(U,t) = F(t); U(0) = U_0, \dot{U}(0) = \dot{U}_0$$

At this stage, let us not inquire into the question of how to arrive at this model. Let us focus on examining questions on numerical integration of this type of equations.

19

Now, we'll now start discussing some issues about integrating equations of motion if we are dealing with nonlinear systems, we will start by discussing some preliminary ideas perhaps later in the course we will address these questions in a more elaborate way. Now why do we get nonlinear systems? The sources of non-linearity in structural dynamics could be due to nonlinear strain displacement relations, or due two non-linear stress-strain relations, or non-linear energy dissipation mechanisms for example friction at joins, aerodynamic damping, hysteresis and so on and so forth. Now for the purpose of illustration what we will do is we will consider a nonlinear equation of motion of this form and at this stage let us not inquire into the question on how to arrive at this finite element model from a given continuum problem, that question will consider later. Now the question that we wish to discuss now is if you were to get this type of equation of motion how will you integrate the equation of motion, what are the issues?

**Newmark's method**

$$\dot{U}_{n+1} = \dot{U}_n + \Delta t\left[(1-\delta)\ddot{U}_n + \delta\ddot{U}_{n+1}\right]\cdots(1)$$

$$U_{n+1} = U_n + \dot{U}_n\Delta t + \Delta t^2\left[\left(\frac{1}{2}-\alpha\right)\ddot{U}_n + \alpha\ddot{U}_{n+1}\right]\cdots(2)$$

$$M\ddot{U}_{n+1} + C\dot{U}_{n+1} + KU_{n+1} + g_{n+1}\left(U_{n+1}\right) = F_{n+1}\cdots(3)$$

$$(2)\Rightarrow \ddot{U}_{n+1} = \frac{1}{\alpha\Delta t^2}\left[U_{n+1} - U_n - \dot{U}_n\Delta t\right] - \frac{1}{\alpha}\left(\frac{1}{2}-\alpha\right)\ddot{U}_n\cdots(4)$$

Substitute (4) in (1) to get

$$\dot{U}_{n+1} = \dot{U}_n + \Delta t\left[(1-\delta)\ddot{U}_n + \frac{\delta}{\alpha\Delta t^2}\left\{U_{n+1} - U_n - \dot{U}_n\Delta t\right\} - \frac{\delta}{\alpha}\left(\frac{1}{2}-\alpha\right)\ddot{U}_n\right]\cdots(5)$$


20

So let's see what happens to Newmark's method, so the representation for velocity and displacement remains the same, and the equilibrium equation at T = N+1 will have now this new term, so upon simplifying these equations from equation 2 I get if I solve for U double dot N+1 I get this, and if I substitute now this and these quantities into the equation of motion or in the first equation I get U dot of N+1 is UN dot plus these terms, and this is what we got earlier also.

$$\dot{U}_{n+1} = \dot{U}_n + \Delta t\left[(1-\delta)\ddot{U}_n + \frac{\delta}{\alpha\Delta t^2}\{U_{n+1} - U_n - \dot{U}_n\Delta t\} - \frac{\delta}{\alpha}\left(\frac{1}{2} - \alpha\right)\ddot{U}_n\right] \cdots (5)$$

Note: (4) and (5) contain the unknown $U_{n+1}$ on the RHS.

Substitute of (4) and (5) into (3) to get

$$\left[\frac{1}{\alpha\Delta t^2}M + \frac{\delta}{\alpha\Delta t}C + K\right]U_{n+1} + g_{n+1}(U_{n+1}) = F_{n+1} + M\left(a_0 U_n + a_2\dot{U}_n + a_3\ddot{U}_n\right)$$

$$+C\left(a_1 U_n + a_4\dot{U}_n + a_5\ddot{U}_n\right) \quad \cdots (6)$$

$$a_0 = \frac{1}{\alpha\Delta t^2}; a_1 = \frac{\delta}{\alpha\Delta t}; a_2 = \frac{1}{\alpha\Delta t}; a_3 = \left(\frac{1}{2\alpha} - 1\right); a_4 = \frac{\delta}{\alpha} - 1;$$

$$a_5 = \frac{\Delta t}{2}\left(\frac{\delta}{\alpha} - 2\right); a_6 = \Delta t(1-\delta); a_7 = \delta\Delta t$$

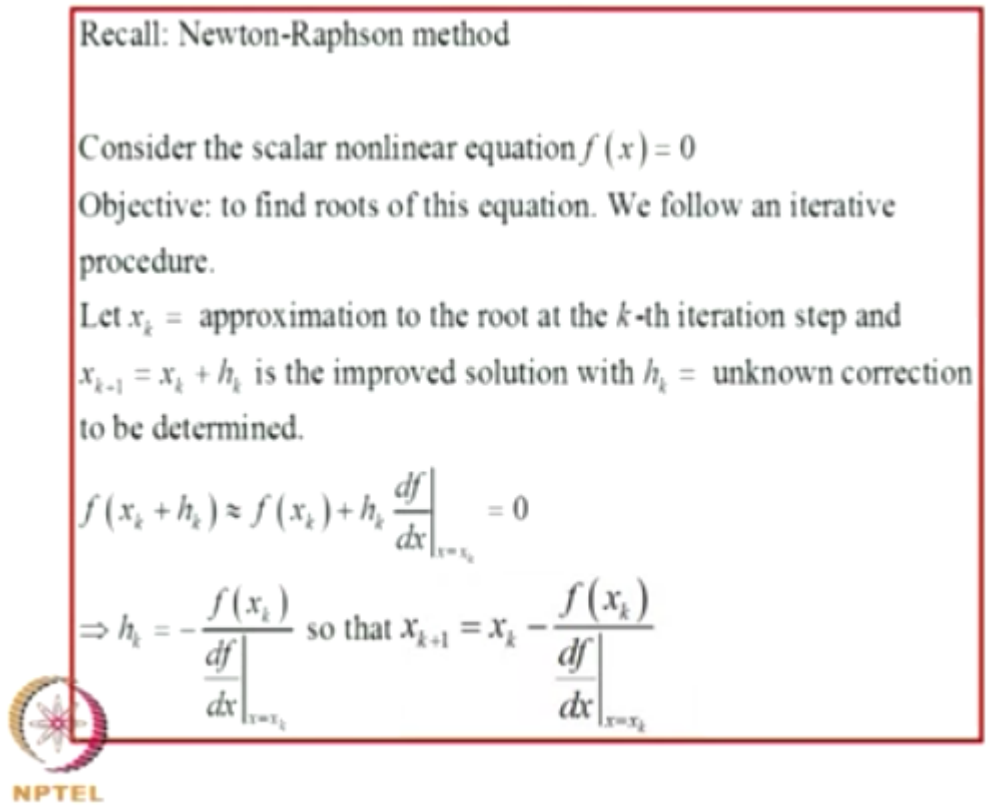
$\Rightarrow$

At every time step we need to solve a set of nonlinear algebraic equations.



Now if we now go back to, we have now acceleration, displacement and I mean velocity and displacement all of that if we now substitute into the governing equation of motion we get this equation for UN+1.

Now this equation also has this term GN+1, UN+1 so at time TN+1 to obtain UN+1 I have to solve this non-linear equation, this is a non-linear algebraic equation. Earlier in the absence of this term we would have inverted this matrix and got the solution, but now these are a set of nonlinear algebraic equations which at every time step we need to solve, okay this is the additional complexity, so how do we tackle this? One of the popular algorithms for this is the




Newton-Raphson method so we will quickly recall what is the, how do we implement that method? Suppose we consider a scalar nonlinear equation F(x) = 0, now our objective is to find roots of this equation, we follow an iterative procedure, so what we do is XK, we consider XK to be approximation to the root at K-th iteration step and XK+1 that is the improved solution is XK plus a correction. Now the objective is to find what this correction should be, so I will consider therefore F(xk+hk) which is since we are truncating this, this will be F(xk) + HK DF/DX at X = XK. Now if XK + HK is a true solution the left hand side would be 0 from which I get HK to be this, so that XK+1 is given by this, so the one quick observation that we need to make is to obtain the rules now I need gradients of function F.

$$x_{k+1} = x_k - \frac{f(x_k)}{\left.\frac{df}{dx}\right|_{x=x_k}}$$

Remarks

- Implicit scheme
- Needs evaluation of the derivative of $f(x)$
- $x_0$ = initial guess on the root
- In integrating equations of motion the initial guess could be the solution from the linear system response or the nonlinear response at the preceding time instant.
- Step size: typically is larger than what is needed for explicit schemes
- Tradeoff: large step size and need to solve nonlinear algebraic equations at every time step




Now we are talking about Newmark's beta method, now if we are going to use Newton-Raphson scheme within that, the scheme would be implicit now it also needs derivatives of the nonlinear functions, and if I take X naught to be initial guess on the root, then we have to make this guess to start the solution. Now in integrating equations of motion the initial guess could be the solution from the linear system response or the nonlinear response at the preceding time instant or some other reasonable approximation. Step size for implicit schemes it typically is larger than what is needed for explicit schemes, the trade-off is large step size and need to solve nonlinear algebraic equations at every time step, when we are talking about implicit schemes.

**Extension to set of nonlinear equations**

Consider the set of nonlinear equations $f_s(x_1, x_2, \cdots x_n) = 0, s = 1, 2, \cdots, n$

Objective: to find roots of this set of equations following an iterative procedure.

Let $x_{ks}, s = 1, 2, \cdots, n =$ approximation to the root at the $k$-th iteration step and $x_{(k+1)s} = x_{ks} + h_{ks}, s = 1, 2, \cdots, n$ are the improved solutions with $h_{ks} =$ unknown corrections to be determined for $s = 1, 2, \cdots, n$.

$$f_s(x_{1k} + h_{k1}, x_{2k} + h_{k2}, \cdots x_{nk} + h_{kn}) = f_s(x_{1k}, x_{2k}, \cdots x_{nk}) + \sum_{j=1}^{n} \left( \frac{\partial f_s}{\partial x_j} \right)_{x = x_k} h_{kj} = 0$$

$s = 1, 2, \cdots, n$

$$\Rightarrow \{h_k\} = -\left[ \left( \frac{\partial f_s}{\partial x_j} \right)_{x = x_k} \right]^{-1} \{f_s(x_{1k}, x_{2k}, \cdots x_{nk})\}$$

$$\Rightarrow \{x_{k+1}\} = \{x_k\} - \left[ \left( \frac{\partial f_s}{\partial x_j} \right)_{x = x_k} \right]^{-1} \{f_s(x_{1k}, x_{2k}, \cdots x_{nk})\}$$

24

Now we will now consider to complete the discussion on Newton-Raphson method a set of nonlinear equations in SN variables, suppose I have FS (X1, X2, XN) = 0 for S = 1, 2, to N, there are N equations in N unknowns. Now how do we solve these equations? So what I do, I take XKS from S = 1, 2 to N to the approximation to the root at the K-th iteration step, and K+1, S is what I need to find. This is XKS + a correction, so I need to find this correction, again we use Taylor's expansion so this is the function FS at the improved at XK+1S and this is given by this. So now by using a first-order Taylor's expansion I get this equation and if this is indeed the true solution the left hand side would be 0, so I get a set of algebraic equations for the unknown increments in X case, so if you solve that I get HK as inverse of this matrix of gradients, this has to be evaluated to implement the method.

**Central difference Method**

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

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

Consider equillibrium at time $t$ and use

$$\dot{U}(t)=\frac{U(t+\Delta t)-U(t-\Delta t)}{2\Delta t}$$

$$\ddot{U}(t)=\frac{1}{\Delta t^2}\left[U(t+\Delta t)-2U(t)+U(t-\Delta t)\right]$$

$$\Rightarrow$$

$$\frac{M}{\Delta t^2}\left[U(t+\Delta t)-2U(t)+U(t-\Delta t)\right]+\frac{C}{2\Delta t}\left[U(t+\Delta t)-U(t-\Delta t)\right]$$

$$+KU(t)+g\left[U(t),t\right]=F(t)$$

$$\left[\frac{M}{\Delta t^2}+\frac{C}{2\Delta t}\right]U(t+\Delta t)=F(t)-\left[K-\frac{2M}{\Delta t^2}\right]U(t)$$

$$-g\left[U(t),t\right]-\left[\frac{M}{\Delta t^2}-\frac{C}{2\Delta t}\right]U(t-\Delta t)$$


25

Now we have discussed the implicit scheme, now let us see if what happens if I now tackle the non-linear equation using explicit scheme, so let us consider the central difference method. So I have this equilibrium equation at time T and these are the initial condition, and this is a time duration for which I want to solve the equation of motion, so as before we approximate U dot(t) by a central difference approximation, and U double dot(t) we get this, this we have done before, now I substitute for these 2 in the governing equation and I get this equation and here the nonlinear term I am writing this equation at T and the nonlinear term is here, the unknown is U(t) + delta T and that is here so on the right hand side I would have had G(U(t)) which would be known to me, so here as you can see we need not solve any nonlinear equations to advance from one step size to the next step size, so this is explicit.

$$\left[\frac{M}{\Delta t^2}+\frac{C}{2\Delta t}\right]U(t+\Delta t)=F(t)-\left[K-\frac{2M}{\Delta t^2}\right]U(t)-g\left[U(t),t\right]-\left[\frac{M}{\Delta t^2}-\frac{C}{2\Delta t}\right]U(t-\Delta t)$$

**Remarks**

- We need not have to solve nonlinear equations at every time step.
- Explicit scheme.
- Requires special starting scheme
- Requires small step size to obtain acceptable accuracy
- The idea of spectral stability does not apply to nonlinear systems
- Energy balance provides a means to get an idea about algorithmic dam
and stability

So we need not have to solve nonlinear equations at every time step, we need to invert this matrix all this is come, these discussions, these issues are common to what we already discussed this is an explicit scheme it requires a special starting scheme, and also for linear systems we already seen that this method is conditionally stable and there is a critical step size which depends on the natural frequency of the, highest natural frequency of the system and we expect that similar requirement also remains valid even when there is a non-linearity may be more stringent requirement, so in any sense, in any case we can expect that to get acceptable solutions from this we need to use a very small step size.

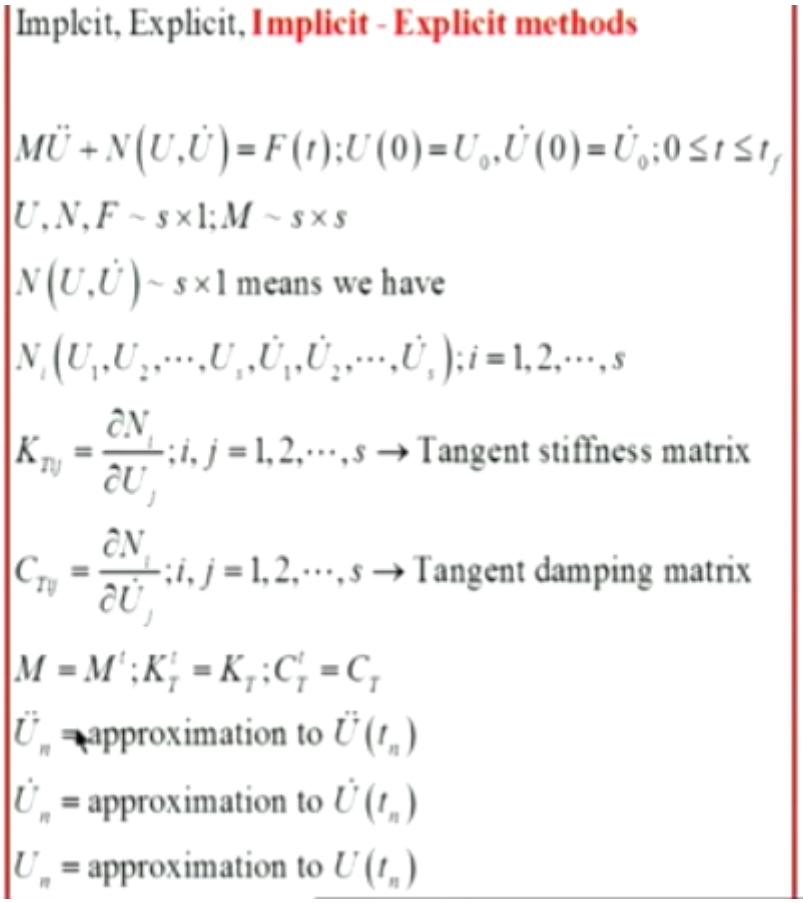
Now the idea of spectral stability does not apply to nonlinear systems, because of the presence of non-linear terms we cannot use the Eigenvalues of the, spectral radius of the amplification matrix of the linear system to infer stability of schemes when applied to nonlinear system, so that issue has to be addressed separately. Now however the concept of energy balance provides a means to get an idea about algorithmic damping and stability, so when you are integrating this we could as well check for energy balance and see whether, how it is behaving, so that may give you some idea about your choice of step size, if energy balance conditions are violated then you are using wrong step size.

**Can we combine explicit and implicit approaches simultaneously in one numerical algorithm?**

T J R Hughes, K S Pister, and R L Taylor, 1979, Implicit-explicit finite elements in nonlinear transient analysis, International Journal of Numerical Methods in Engineering, 17/18, 159-182.

T J R Hughes, R E Stephenson, 1981, Convergence of implicit-explicit algorithms in nonlinear transient analysis, International Journal of Engineering Science, 19, 295-302.

Now we have talked about explicit scheme and implicit scheme. Implicit schemes permits you to use larger step size, but at every time step you have to solve non-linear algebraic equation, explicit schemes there is no need to solve nonlinear algebraic equations but they require small step size. Now the question is can we combine explicit and implicit approaches simultaneously in one numerical algorithm, okay? For example if we have a structural system in which there is local non-linearity, so where there is non-linearity you could use explicit schemes and where there is linearity you can parts of the structures are linear you could use implicit scheme, so that large step sizes could be used for certain parts of the system and explicit schemes the smaller step size can be used for certain other parts, so these ideas have been discussed in papers by Hughes, Pister, and Taylor and there is another paper I have given the reference, so what I will do now is I will quickly outline what the main issue is in this discussion.




Now we want to discuss now implicit, explicit and implicit explicit methods, actually we need to discuss how to combine implicit and explicit approaches in a single, in a simultaneous linear same algorithm, so we will review that we'll start with an implicit formulation and then we will look at explicit formulation in some detail and then see what are the issues when we combine the two. So let's consider the equation of motion MU double dot + N(U,U dot) = F(t), this N is for nonlinear terms, so with certain initial condition and these are time duration of interest, so this U, N and F are S cross 1 matrices, M is S cross S mass matrix, it could be diagonal and N is S cross 1 means actually what we have, we have to understand that there are S number of functions, each one of which is function of U1, U2, US and U1 dot, U2 dot, and US dot, so I runs from 1, 2, S.

Now we define what is known as tangent stiffness matrix where we differentiate the nonlinear term with respect to UJ, so i-th function d function differentiated with respect to J will give me the IJ-th element of the tangent stiffness matrix. Similarly we talk about a tangent damping matrix where I differentiate NI with respect to UJ dot and I get CT IJ, so this is tangent damping matrix. Now we can see that mass matrix is symmetric, and tangent stiffness and tangent damping matrices are also symmetric, now we denote by UN double dot ,UN dot and UN, the approximations to U double dot(tn), and U dot(tn) and U(tn) respectively, so these are exact which are not known these are the approximations.

Implicit scheme: Newmark's method

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

$$\Rightarrow M\ddot{U}_{n+1}+N\left[U_{n+1},\dot{U}_{n+1}\right]=F_{n+1}$$

$$U_{n+1}=U_n+\dot{U}_n\Delta t+\Delta t^2\left[\left(\frac{1}{2}-\alpha\right)\ddot{U}_n+\alpha\ddot{U}_{n+1}\right]$$

$$\Rightarrow U_{n+1}=\tilde{U}_{n+1}+\Delta t^2\alpha\ddot{U}_{n+1}$$

$$\dot{U}_{n+1}=\dot{U}_n+\Delta t\left[(1-\delta)\ddot{U}_n+\delta\ddot{U}_{n+1}\right]$$

$$\Rightarrow \dot{U}_{n+1}=\dot{\tilde{U}}_{n+1}+\Delta t\delta\ddot{U}_{n+1}$$

$$\left.\begin{aligned}\tilde{U}_{n+1}&=U_n+\dot{U}_n\Delta t+\Delta t^2\left(\frac{1}{2}-\alpha\right)\ddot{U}_n\\ \dot{\tilde{U}}_{n+1}&=\dot{U}_n+\Delta t(1-\delta)\ddot{U}_n\end{aligned}\right\}\text{Predictor values}$$

$$\left.\begin{aligned}U_{n+1}&=\tilde{U}_{n+1}+\Delta t^2\alpha\ddot{U}_{n+1}\\ \dot{U}_{n+1}&=\dot{\tilde{U}}_{n+1}+\Delta t\delta\ddot{U}_{n+1}\end{aligned}\right\}\text{Corrector values}$$




Now let's start by discussing the solution for an implicit scheme, let's use in Newmark's method for illustration, it could be other method as well, it could be HHT alpha method or generalized alpha method, it could be anything else, but for purpose of discussion we will stick to Newmark's method. So the equilibrium equation at time T is given by this, and at time N+1 the approximate equation will be this. Now the Newmark method we assume that displacement is given by this where alpha is an algorithmic parameter, and similarly what I will do now is I will collect the term U double dot N+1 and delta T square alpha this is one term, this term makes the algorithm implicit so all other terms which are at N are lumped in this U tilde N+1. Similarly for velocity U dot of N+1 is given by this, delta is algorithmic parameter, again here I will collect all the terms involving velocities and accelerations at TN and call this as U tilde dot, N+1 and the other term which makes the algorithm implicit I will collect it here, so this tilde are U tilde N+1 is this, and U tilde dot N+1 is given by this, we call these values as predictor values, as you enter any time you would be knowing this at any given time step you will be knowing this from your calculations at the preceding steps. Now this is what we should find, these are the corrector values.

(1) Set $t = 0; n = 0$.

$$M\ddot{U}_0 + N\left[U_0, \dot{U}_0\right] = F_0 \Rightarrow \ddot{U}_0 = M^{-1}\left\{F_0 - N\left[U_0, \dot{U}_0\right]\right\}$$

(2) Predictor values

$$\tilde{U}_{n+1} = U_n + \dot{U}_n \Delta t + \Delta t^2 \left(\frac{1}{2} - \alpha\right)\ddot{U}_n$$

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

(3) Corrector step

$$U_{n+1} = \tilde{U}_{n+1} + \Delta t^2 \alpha \ddot{U}_{n+1} = \tilde{U}_{n+1} + \Delta t^2 \alpha \left\{M^{-1}\left\{F_0 - N\left[U_{n+1}, \dot{U}_{n+1}\right]\right\}\right\}$$

$$\dot{U}_{n+1} = \tilde{\dot{U}}_{n+1} + \Delta t \delta \ddot{U}_{n+1} = \tilde{\dot{U}}_{n+1} + \Delta t \delta \left\{M^{-1}\left\{F_0 - N\left[U_{n+1}, \dot{U}_{n+1}\right]\right\}\right\}$$

These two equations for a set of nonlinear equations for the unknowns $U_{n+1}$ & $\dot{U}_{n+1}$. We use Newton-Raphson's method with predictors as starting solution.

30

Now how do we implement this method? We set T = 0, and N = 0, so T = 0 the initial displacement and velocity are known, and using the equation of motion I derive the initial acceleration. Now predictor value is, now N is 0 therefore UN tilde will be U naught + U naught dot delta T + delta T square U naught double dot, this into this term, so all these terms are known. Similarly U tilde dot 1 because N is 0 is U dot naught + delta T 1 - delta U double dot naught that is known, so that's what I meant predictor values will be known when you enter this step.

Now the corrector steps would be given by this, now for U double dot N+1 I can use the equation of motion and write this terms there, okay, this term, this M inverse of this. Similarly U dot N+1 is the predictor term + this term, now you see here the unknowns are UN+1 and U dot N+1, and on the right hand side the unknowns are contained in this nonlinear terms, so actually these two equation therefore form a set of nonlinear equations for the unknowns UN+1 and U dot N+1, we use Newton-Raphson's method with predictors as starting solution, we use this tilde as quantities with tilde relations as the initial guess, and use Newton Raphson, so how does that work out? So I have the first equation is given by this, I have taken the terms on the

$$U_{n+1} - \tilde{U}_{n+1} + \Delta t^2 \alpha \left\{ M^{-1} \left\{ F_0 - N\left[ U_{n+1}, \dot{U}_{n+1} \right] \right\} \right\} = 0 \Rightarrow G\left( U_{n+1}, \dot{U}_{n+1} \right) = 0$$

$$\dot{U}_{n+1} - \dot{\tilde{U}}_{n+1} + \Delta t \delta \left\{ M^{-1} \left\{ F_0 - N\left[ U_{n+1}, \dot{U}_{n+1} \right] \right\} \right\} = 0 \Rightarrow H\left( U_{n+1}, \dot{U}_{n+1} \right) = 0$$

Newton Raphson Solution

$$U_{n+1}^{i+1} = U_{n+1}^{i} + \Delta_i$$

$$\dot{U}_{n+1}^{i+1} = \dot{U}_{n+1}^{i} + \dot{\Delta}_i$$

$$G\left( U_{n+1}^{i} + \Delta_i, \dot{U}_{n+1}^{i} + \dot{\Delta}_i \right) = 0 \Rightarrow$$

$$G\left( U_{n+1}^{i}, \dot{U}_{n+1}^{i} \right) + \sum_{j=1}^{s} \left( \frac{\partial G}{\partial U_{n+1}} \right)_{\substack{U_{n+1} = U_{n+1}^{i} \\ \dot{U}_{n+1} = \dot{U}_{n+1}^{i}}} \Delta_{ij} + \sum_{j=1}^{s} \left( \frac{\partial G}{\partial \dot{U}_{n+1}} \right)_{\substack{U_{n+1} = U_{n+1}^{i} \\ \dot{U}_{n+1} = \dot{U}_{n+1}^{i}}} \dot{\Delta}_{ij} = 0$$

$$H\left( U_{n+1}^{i} + \Delta_i, \dot{U}_{n+1}^{i} + \dot{\Delta}_i \right) = 0 \Rightarrow$$

$$H\left( U_{n+1}^{i}, \dot{U}_{n+1}^{i} \right) + \sum_{j=1}^{s} \left( \frac{\partial H}{\partial U_{n+1}} \right)_{\substack{U_{n+1} = U_{n+1}^{i} \\ \dot{U}_{n+1} = \dot{U}_{n+1}^{i}}} \Delta_{ij} + \sum_{j=1}^{s} \left( \frac{\partial H}{\partial \dot{U}_{n+1}} \right)_{\substack{U_{n+1} = U_{n+1}^{i} \\ \dot{U}_{n+1} = \dot{U}_{n+1}^{i}}} \dot{\Delta}_{ij} = 0$$



right hand side to left hand side and I equate it to 0, let us call it as some capital G(UN+1, U dot N+1) is 0. The second equation for velocity leads to this equation. Now these two are pairs of nonlinear equations in UN+1 and U dot N+1, so there are 2S number of unknowns, and there are 2S number of equations.

Now Newton-Raphson solution now we have to set up an iteration, and that iteration count is I, so I will first give a few mathematical details and then we'll return to the algorithmic implementation, so at I-th step if this is the guess and delta I is the correction, then this is the improved solution. So similarly on velocity this is the improved solution, now you substitute this into these two equations whatever we discussed till now use first-order Taylor's expansion I get this expression, so what are unknowns here? This delta IJ and delta dot IJ are the unknowns which we need to find. Similarly using Taylor's expansion on H, I get this equation, so I have now, there are 2S number of unknowns and 2S number of linear algebraic equations they can be cast in this form so the work involves evaluation of this matrix and inversion, so you have this.

$$\Rightarrow \begin{Bmatrix} \Delta_i \\ \dot{\Delta}_i \end{Bmatrix} = \begin{bmatrix} \left(\frac{\partial G}{\partial U_{n+1}}\right)_{\substack{U_{n+1}=U^i_{n+1} \\ \dot{U}_{n+1}=\dot{U}^i_{n+1}}} & \left(\frac{\partial G}{\partial \dot{U}_{n+1}}\right)_{\substack{U_{n+1}=U^i_{n+1} \\ \dot{U}_{n+1}=\dot{U}^i_{n+1}}} \\ \left(\frac{\partial H}{\partial U_{n+1}}\right)_{\substack{U_{n+1}=U^i_{n+1} \\ \dot{U}_{n+1}=\dot{U}^i_{n+1}}} & \left(\frac{\partial H}{\partial \dot{U}_{n+1}}\right)_{\substack{U_{n+1}=U^i_{n+1} \\ \dot{U}_{n+1}=\dot{U}^i_{n+1}}} \end{bmatrix}^{-1} \begin{Bmatrix} G\left(U^i_{n+1},\dot{U}^i_{n+1}\right) \\ H\left(U^i_{n+1},\dot{U}^i_{n+1}\right) \end{Bmatrix}$$

$$G\left(U_{n+1},\dot{U}_{n+1}\right) = U_{n+1} - \tilde{U}_{n+1} + \Delta t^2 \alpha \left\{ M^{-1} \left\{ F_0 - N\left[U_{n+1},\dot{U}_{n+1}\right] \right\} \right\}$$

$$H\left(U_{n+1},\dot{U}_{n+1}\right) = \dot{U}_{n+1} - \dot{\tilde{U}}_{n+1} + \Delta t \delta \left\{ M^{-1} \left\{ F_0 - N\left[U_{n+1},\dot{U}_{n+1}\right] \right\} \right\}$$

$$\frac{\partial G}{\partial U_{n+1}} = 1 - \Delta t^2 \alpha M^{-1} \frac{\partial N}{\partial U_{n+1}} = 1 - \Delta t^2 \alpha M^{-1} K_T$$

$$\frac{\partial G}{\partial \dot{U}_{n+1}} = -\Delta t^2 \alpha M^{-1} \frac{\partial N}{\partial \dot{U}_{n+1}} = -\Delta t^2 \alpha M^{-1} C_T$$

$$\frac{\partial H}{\partial U_{n+1}} = -\Delta t \delta M^{-1} \frac{\partial N}{\partial U_{n+1}} = -\Delta t \delta M^{-1} K_T$$

$$\frac{\partial H}{\partial \dot{U}_{n+1}} = 1 - \Delta t \delta M^{-1} \frac{\partial N}{\partial \dot{U}_{n+1}} = 1 - \Delta t \delta M^{-1} C_T$$




Now G function is consequently this, and H function, where G and H are this, I am just repeating for sake of completion here. Now I need these gradients dou G/dou U, dou G/dou dot, dou H/dou U and dou H/U dot so they can be evaluated from the Newton-Raphson, sorry Newmark beta, Newmark's as a model, and we get these quantities in terms of tangent stiffness matrix and tangent damping matrix as shown here. So in terms of, we can write now this matrix in terms of KT and CT which we will do now.

(4) Newton Raphson Steps

(4.1) Initialize

$i = 0; U_{n+1}^{i} = \tilde{U}_{n+1}; \dot{U}_{n+1}^{i} = \tilde{\dot{U}}_{n+1}$

(4.2) Iterate

$$\begin{Bmatrix} U_{n+1}^{i+1} \\ \dot{U}_{n+1}^{i+1} \end{Bmatrix} = \begin{Bmatrix} U_{n+1}^{i} \\ \dot{U}_{n+1}^{i} \end{Bmatrix} + \begin{bmatrix} \left(1 - \Delta t^2 \alpha M^{-1} K_T\right)_{\substack{U_{n+1}=U_{n+1}^{i} \\ \dot{U}_{n+1}=\dot{U}_{n+1}^{i}}} & \left(-\Delta t^2 \alpha M^{-1} C_T\right)_{\substack{U_{n+1}=U_{n+1}^{i} \\ \dot{U}_{n+1}=\dot{U}_{n+1}^{i}}} \\ \left(-\Delta t \delta M^{-1} K_T\right)_{\substack{U_{n+1}=U_{n+1}^{i} \\ \dot{U}_{n+1}=\dot{U}_{n+1}^{i}}} & \left(1 - \Delta t \delta M^{-1} C_T\right)_{\substack{U_{n+1}=U_{n+1}^{i} \\ \dot{U}_{n+1}=\dot{U}_{n+1}^{i}}} \end{bmatrix}^{-1} \begin{Bmatrix} G\left(U_{n+1}^{i}, \dot{U}_{n+1}^{i}\right) \\ H\left(U_{n+1}^{i}, \dot{U}_{n+1}^{i}\right) \end{Bmatrix}$$

$i \rightarrow i+1$

If $\left|U_{n+1}^{i+1} - U_{n+1}^{i}\right| \le \varepsilon_1$ & $\left|\dot{U}_{n+1}^{i+1} - \dot{U}_{n+1}^{i}\right| \le \varepsilon_2$ exit to (5)

Else go to (4.2) and continue with iterations

(5) $U_{n+1} = U_{n+1}^{i}$ & $\dot{U}_{n+1} = \dot{U}_{n+1}^{i}$

(6) $M\ddot{U}_{n+1} + N\left[U_{n+1}, \dot{U}_{n+1}\right] = F_{n+1} \Rightarrow \ddot{U}_{n+1} = M^{-1}\left\{F_{n+1} - N\left[U_{n+1}, \dot{U}_{n+1}\right]\right\}$

(7) $n \rightarrow n+1$; if $n\Delta t \ge t_f$ exit; else go to (2).

NPTEL

33

So let us now return to you know we started with the 3 steps, we completed the predictor step and we are now formulating the corrector steps, so corrector steps now involves Newton-Raphson steps, so a 4.1 we initialize, I = 0 that's a iteration and these are the initial guesses which are the predictors which we have already derived. Now iterations begin so I use this relation which is Newton-Raphson relation, and increment I and then I will see whether this algorithm has converged by using epsilon 1 and epsilon 2 as measures of you know convergence, if this is satisfied we exit out of this iteration loop, otherwise we go to 4.2 with, continue with these iterations. Now if the convergence requirements are met we will be here and I will assign now to the displacement at N+1 and velocity at N+1 these converge quantities, then I have to compute U double dot N+1 for which again I use the equation of motion.

Now I increment N the time marching and if I cross the final step size I exit, otherwise I go to step 2 which is that? We again start with predictor for the next step, okay? So this is implementation for, implementation of Newmark's method for a nonlinear system, this is an implicit scheme because as we have seen average time step we are solving a set of non-linear equations.

**Explicit scheme : Predictor - Corrector method**

Idea: introduce the equation

$$M\ddot{U}_{n+1} + N\left[\tilde{U}_{n+1}, \dot{\tilde{U}}_{n+1}\right] = F_{n+1}$$

Use this equation in implementing the corrector step.

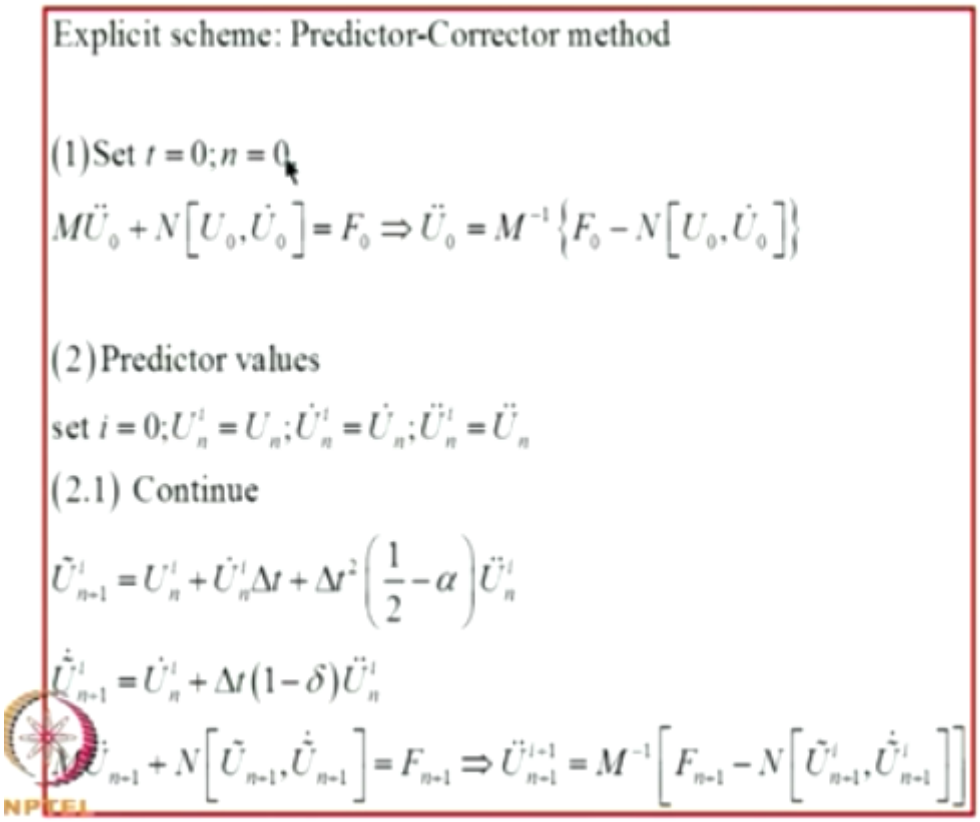
This avoids the solution of the set of simultaneous nonlinear equations at every time step.

The method becomes explicit.

NPTEL

34

Now we look at the same problem formulation using explicit scheme, now what we do is we introduce this additional equation, see we got nonlinear equations in the preceding step because for U double dot N+1 I replaced it by, if you see here carefully for U double dot I have N+1, I am replacing this by F naught – N(UN+1, U dot N+1) and we are solving these nonlinear equations, if instead of that for these terms inside this bracket if I use the predictors, then this becomes explicit, okay so that leads us to the next scheme, the idea is we will use this as, we introduce this equation MU double dot N+1 as N U tilde N+1, U dot this also tilde, now use this equation implementing the corrector step, that means at the stage of implementing the corrector step we will not use, we need not have to solve a set of nonlinear algebraic equation, but we still need to iterate, there will be multiple passes that will be needed and iteration will be needed, this avoids the solution to the set of simultaneous nonlinear equations at every time step and the method becomes explicit.





So how does it work? You start with T = 0, you have initial displacement and velocity use equation of motion get initial acceleration, then you use predictor values you set I = 0, and your UN(i) is U naught, U1 dot(i) is U naught dot and U double dot(i) is U double dot naught when you are at N = 0. Next I need U tilde of I N+1, this is the predictor, this we have, this is for displacement, this is for velocity. Now after having found out I want now an improved estimate of U double dot N+1, to do that I will use the equation of motion where U and U dot are replaced by the predictor values, so I get consequently U double dot I+1, N+1 is given by this, this is an explicit equation there is no nonlinear equation solving here, so I get acceleration

$$U_{n+1}^{i+1} = \tilde{U}_{n+1}^{i} + \Delta t^2 \alpha \ddot{U}_{n+1}^{i+1}$$

$$\dot{U}_{n+1}^{i+1} = \tilde{\dot{U}}_{n+1}^{i} + \Delta t \delta \ddot{U}_{n+1}^{i+1}$$

If $\left|U_{n+1}^{i+1} - U_{n+1}^{i}\right| \le \varepsilon_1$ & $\left|\dot{U}_{n+1}^{i+1} - \dot{U}_{n+1}^{i}\right| \le \varepsilon_2$ exit and go to (3)

Else $i \to i+1$

Go to (2.1)

$$(3) U_{n+1} = U_{n+1}^{i}, \dot{U}_{n+1} = \dot{U}_{n+1}^{i} \;\&\; \ddot{U}_{n+1} = M^{-1}\left\{F_{n+1} - N\left[U_{n+1}, \dot{U}_{n+1}\right]\right\}$$

$(4) n \to n+1$. If $n\Delta t \ge t_f$ exit and stop.

Else go to $(2)$




from which I will again improve upon the displacement and velocity and check whether the convergence has occurred, if it has occurred we will exit the iteration loop otherwise we iterate once again with an improved, the predictor values are now replaced by these improved values.

So we if we exit after satisfactory convergence I get UN+1 which is the, displacement will be what has converged here UI N+1, similarly velocity and acceleration are obtained, and here I will use the, not the predictors but the converge solutions to find acceleration. Then I will increment time and I will exit if we have cross the final time instant, otherwise we go back and continue with the algorithm, so this is an explicit scheme, it involves iterations, multiple passes but there is no nonlinear equation solving okay. Now the idea is the question that we are trying to ask is now can we combine these 2 approaches?

**Implicit - Explicit scheme : Mesh partitioning**

Use explicit and implicit concepts simultaneously in one algorithm.

Idea: Divide the finite elements into "implicit" & "explicit" sets.

Let superscripts $I$ and $E$ denote the two sets respectively.

$M^I, N^I, F^I$ = assembled mass, internal force, and external force for the implicit elements

$M^E, N^E, F^E$ = assembled mass, internal force, and external force for the explicit elements

$$\Rightarrow$$

$$\left(M^I + M^E\right)\ddot{U} + N^I\left[U, \dot{U}\right] + N^E\left[U, \dot{U}\right] = F^I(t) + F^E(t)$$

$$\Rightarrow$$

$$\left(M^I + M^E\right)\ddot{U}_{n+1} + N^I\left[U_{n+1}, \dot{U}_{n+1}\right] + N^E\left[\tilde{U}_{n+1}, \dot{\tilde{U}}_{n+1}\right] = F^I(t) + F^E(t)$$

NPTEL



So the question is can we use explicit and implicit concepts simultaneously in one algorithm, the basic idea is we divide the finite elements into implicit and explicit sets, that means the mesh, finite element mesh part of the mesh is designated as implicit, part of the mesh is designated as explicit, so where you expect for instance local nonlinearities you could or stress concentrations or things like that you use finer mesh or you use explicit mesh, where you expect your understand, I mean linear behavior or things like that you could use implicit mesh.

Now let us yeah the superscripts I and E denote the two sets respectively, that is implicit and explicit, so associated with this partitioning I will have a MI, NI, FI as assembled mass, internal force, and an external force for the implicit elements, and similarly ME, NE, FE as assembled mass, internal force and external force for the explicit elements. So the equation of motion now can be written as (MI+ME) U double dot + NI (U,U dot) + NE (U,U dot) is FI(t) FE(t), there is no, this is simply rewriting the equation of motion without making any approximation. The approximation is now is displayed here, for implicit part of the mesh I will use the current states UN+ 1 and U dot N+1, for the explicit parts I will use the predictors, okay. So the idea is when you solve for the nonlinear equations you will be using NI naught N okay, and this will be, the structure of the matrices that you encounter and there banded ness features etc is controlled by NI not by N, and NE makes no contribution to that in your solution to nonlinear equations, so the effort in solving the nonlinear equations can come down substantially, so these are the

$$\left.\begin{aligned}\tilde{U}_{n+1} &= U_n + \dot{U}_n \Delta t + \Delta t^2 \left(\frac{1}{2} - \alpha\right) \ddot{U}_n \\ \dot{\tilde{U}}_{n+1} &= \dot{U}_n + \Delta t (1-\delta) \ddot{U}_n\end{aligned}\right\} \text{Predictor values}$$

$$U_{n+1} = \tilde{U}_{n+1} + \Delta t^2 \alpha \ddot{U}_{n+1}$$

$$= \tilde{U}_{n+1} + \Delta t^2 \alpha \left\{ M^{-1} \left\{ F_0 - N^I \left[ U_{n+1}, \dot{U}_{n+1} \right] - N^E \left[ \tilde{U}_{n+1}, \dot{\tilde{U}}_{n+1} \right] \right\} \right\}$$

$$\dot{U}_{n+1} = \dot{\tilde{U}}_{n+1} + \Delta t \delta \ddot{U}_{n+1}$$

$$= \dot{\tilde{U}}_{n+1} + \Delta t \delta \left\{ M^{-1} \left\{ F_0 - N^I \left[ U_{n+1}, \dot{U}_{n+1} \right] - N^E \left[ \tilde{U}_{n+1}, \dot{\tilde{U}}_{n+1} \right] \right\} \right\}$$

The tangent stiffness and tangent damping matrices are evaluated as

$$K_T^I = \frac{\partial N^I}{\partial U} \;\&\; C_T^I = \frac{\partial N^I}{\partial \dot{U}}$$

The band profile of these matrices would correspond only to the connectivity of the implicit elements.



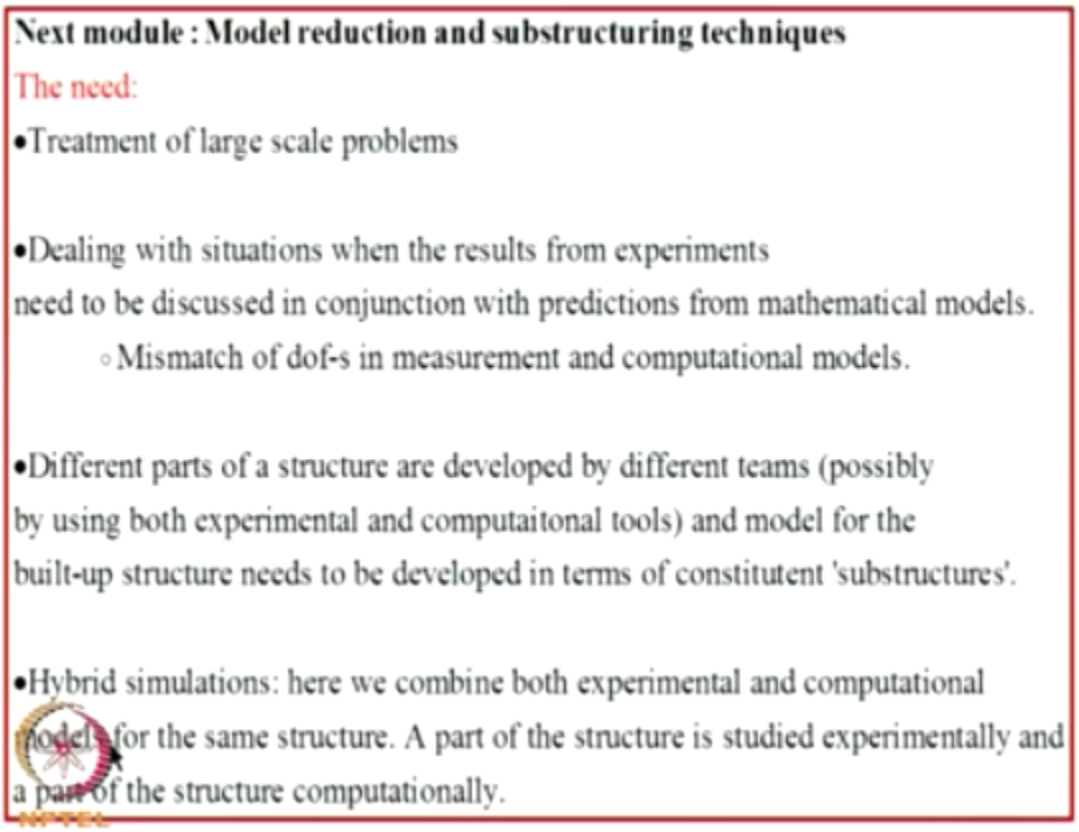
predictor values as before, and I get UN+1 is the predictor plus this, which now I will write for U double dot N+1, in implicit scheme I wrote simply the governing equation here, in the explicit scheme I used the predictor equation, so here what I am doing I am splitting for implicit elements I am writing NI with current UN+1 and U dot N+1 retained as it is, for explicit elements I am using the predictors. So the unknown nonlinear equation, effort to solve nonlinear equation is now related to this NI, so similarly I get an equation for U dot N+1 which is again written here. The tangent stiffness and tangent damping matrices are now evaluated only with respect to NI, you don't need the tangent KT and CT for complete N vector, so you need only for NI, and only for NI, so consequently the band profile of these matrices would correspond only to the connectivity of the implicit elements, so they can be quite sparse and well-structured and maybe easier to solve.

Remark

- We have discussed the implicit, explicit, and implicit-explicit methods in the context of Newmark's method. Similar discussion can also be developed for other schemes such as HHT-$\alpha$ method and generalized $\alpha$ methods.
- The partitioning of the mesh into explicit and implicit elements can also be interptreted as splitting the operator into explicit and implicit parts.



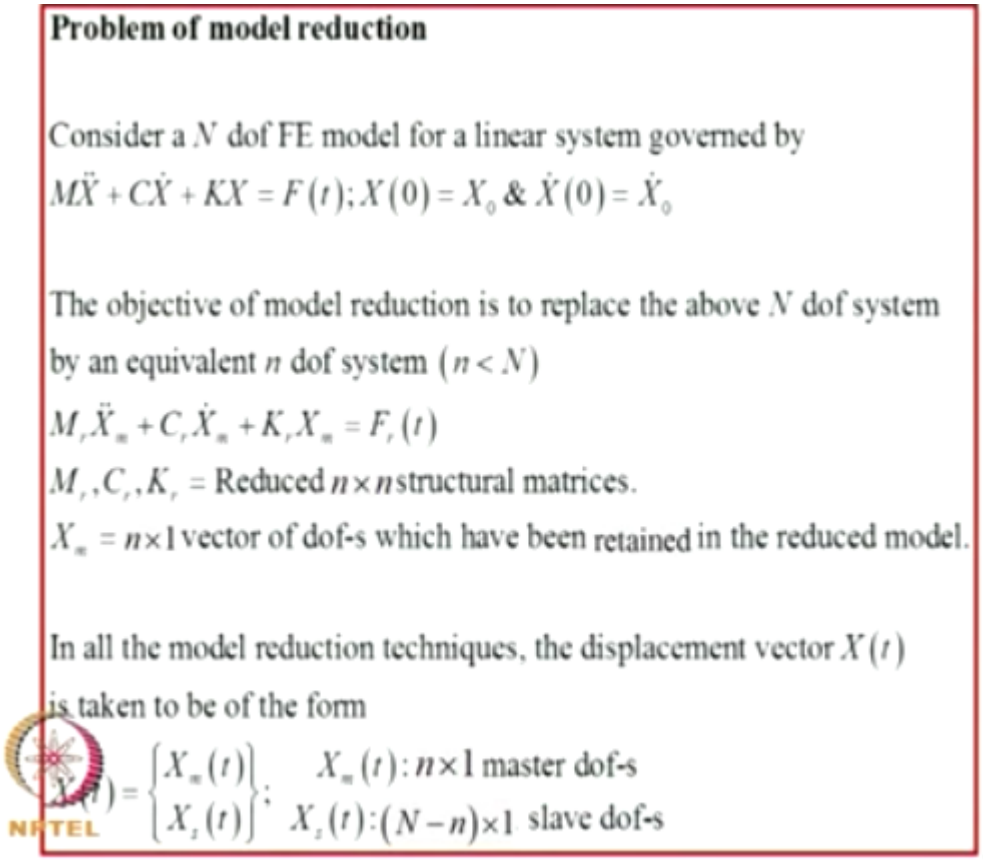

Now I will not get into the details of this implementation but we can make some observations, now what we have done is we have discussed the implicit, explicit, and implicit-explicit methods in the context of Newmark's method. Now same discussion or similar discussion can also be developed for other schemes such as HHT alpha and generalized alpha methods, so the predictor equations that you derive for HHT alpha should originate from HHT alpha representation, here the predictor equation you know was from the Newmark beta method, Newmark's method, so that's one thing that we have to bear in mind. Now the partitioning of the mesh into explicit and implicit elements can also be interpreted as splitting the operator into explicit and implicit parts, so this is so-called operator splitting implicit-explicit method.




Now what I wish to do in the remaining few minutes of this lecture is to give a brief introduction to what are the issues that we would like to address in the next module. The next module of this course is on model reduction and sub-structuring techniques. Why do we need that? The need arises because of, for example treatment of large scale problems I may like to reduce the size of the problem. Now the need also arises whenever you have to deal with situations when results from experiments need to be discussed in conjunction with predictions from mathematical models. For example if you have an instrumented structure there, if you say that if you have used say S number of sensors, the experimental model will have S degrees of freedom if we call the number of sensors as degrees of freedom, it has S degrees of freedom, but the computational model for the same structure can have different degrees of freedom, typically the size of a mathematical model could be fairly large compared with number of sensors that we use in experimental part, so there is a mismatch of degrees of freedom in computational model and experimental model, so how do we deal with that? Or the other situation is different parts of a structure could be developed by different teams, possibly working independently, working independently and possibly by using both experimental and computational tools, for example this type of issues can arise in dealing with automotive systems or space vehicles and things like that where different parts of the structure can be designed and developed by different teams, so each team will make a valid finite element model for the particular sub-structure that they are dealing with and they may also conduct experiments on each of the sub-structures that they are studying.

And finally when the structure is assembled then computational model need to be assembled, the experimental model needs to be assembled, so what are the issues? So these issues take us

to a discussion on sub-structuring methods. Finally another context in which these issues will become important is in the context of what are known as hybrid simulations, here the word hybrid means in testing of structures under dynamic loads we combine both experimental and computational methods, that is why they are called hybrid simulation. Hybrid simulations are essentially experimental testing techniques to qualify structures for specified dynamic loads, for example earthquake qualification testing of a structure, here we combine both experimental and computational models for the same structure and the same team works on both of them, so a part of the structure is studied experimentally, and a part of the structure studied computationally. So how do we combine in the mathematical model that we use in our computational model with the part of the structure which is actually studied experimentally and for which no mathematical model is done, so this takes us to different types of sub-structuring problems.




Now I'll briefly introduce the problem of model reduction, then we'll take up discussions in subsequent lectures. Now suppose we consider a N degree of freedom finite element model for a linear system, say the linear system is governed by MX double dot + CX + KX is F(t) with some initial conditions. What is the problem of model reduction here? The objective of model reduction is to replace this N degree of freedom system by an equivalent lower order system, a lower case N degree of freedom system, where the reduced system will have much lesser number of degrees of freedom than the larger system, so the objective is to derive the equation of motion for another system which has less number of degrees of freedom as shown here, and these two should be related this MR, CR, KR are the reduced N cross N structural matrix. XM is N cross 1 vector of degrees of freedom which have been retained in the reduced model, in all the model reduction techniques the displacement vector X(t) is related to, I mean is partitioned

as XM(t) and XS(t), XM(t) are the degrees of freedom that we want to retain in the reduced model XS(t) are the degrees of freedom which we would like to eliminate, so this M and S refer to the subscripts M refers to master SS slave, so those degrees of freedom which are retained are called master degrees of freedom, those degrees of freedom which are eliminated are called slave degrees of freedom. So XM(t) is N cross 1, master degrees of freedom and XS(t) is N - N slave degrees of freedom.

We represent

$$X(t) = \begin{Bmatrix} X_m(t) \\ X_s(t) \end{Bmatrix} = \Psi X_m(t)$$

where $\Psi$ is an $N \times n$ transformation matrix.

$\Rightarrow$

$$M\Psi\ddot{X}_m(t) + C\Psi\dot{X}_m(t) + K\Psi X_m(t) = F(t)$$

$\Rightarrow$

$$\Psi^t M\Psi\ddot{X}_m(t) + \Psi^t C\Psi\dot{X}_m(t) + \Psi^t K\Psi X_m(t) = \Psi^t F(t)$$

$\Rightarrow$

$$M_r\ddot{X}_m + C_r\dot{X}_m + K_r X_m = F_r(t)$$

$M_r = \Psi^t M\Psi$ = Reduced mass matrix; $M_r^t = M_r$

$C_r = \Psi^t C\Psi$ = Reduced damping matrix; $C_r^t = C_r$

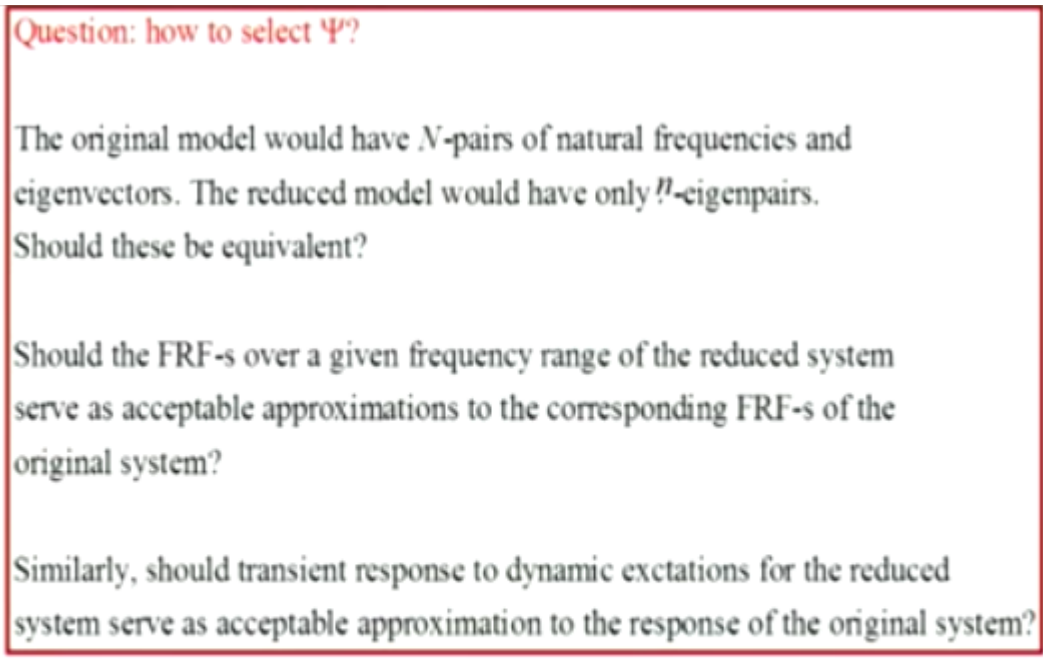
$K_r = \Psi^t K\Psi$ = Reduced stiffness matrix; $K_r^t = K_r$

$F_r(t) = \Psi^t F(t)$ = Reduced force vector

Question: how to select $\Psi$?

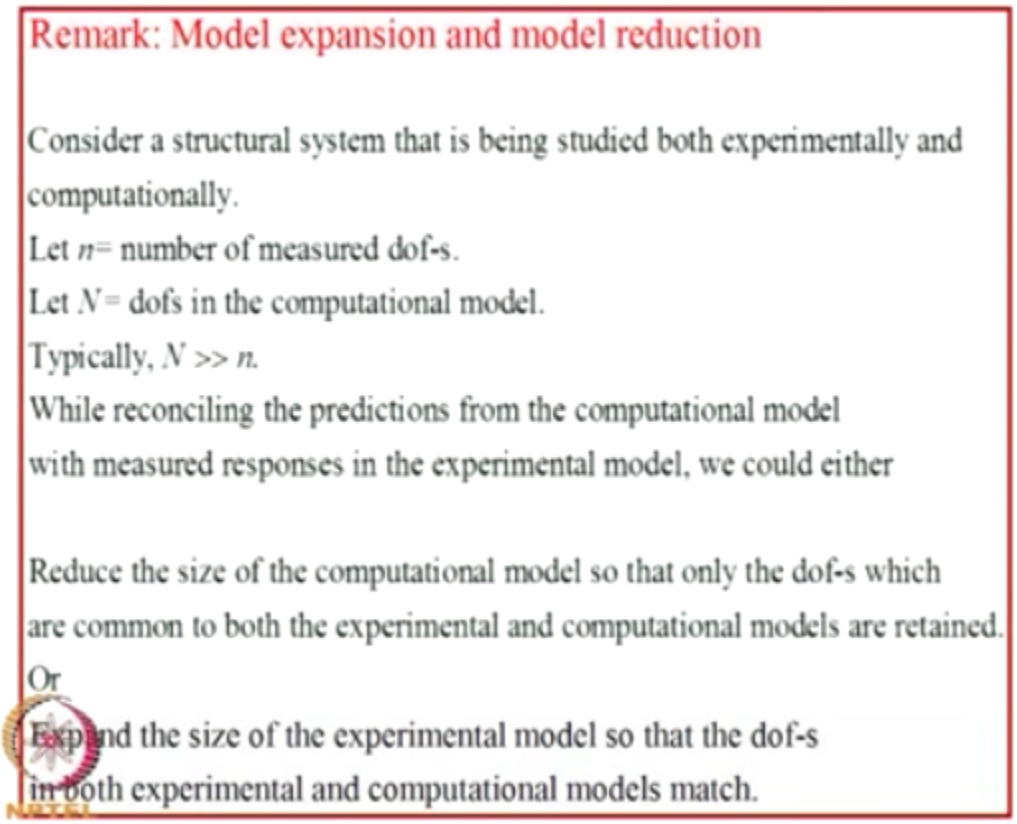
42

Now we represent X(t) as some sai into XM(t), okay, this is how we eliminate the slave degrees of freedom, where sai is an N cross M transformation matrix, so we substitute this into the governing equation I have M sai XM double dot + C sai XM dot + and for X I am writing sai into XM, so this is M sai XM double dot + C sai XM dot + K sai XM is F(t), I'll pre multiply by sai transpose, so I get this equation. So now I will call sai transpose M sai as MR which is a reduced mass matrix, and CR which is sai transpose C sai is the reduced damping matrix, KR is sai transpose K sai which is a reduced stiffness matrix, you can quickly verify that these matrices are all symmetric, and FR(t) is sai transpose F(t) which is a reduced force vector.




Now the question is how do we select sai? Okay, that is the main question. When we want to, when we reduce the size of the model for example the original model would have N pairs of natural frequencies and Eigen vectors, the reduced model will have lowercase N Eigen pairs, should these be equivalent? For example 100 degrees of freedom is represented in terms of 10 degrees of freedom model, so the 100 degrees of freedom will have 100 natural frequencies possible, and a 10 degree of freedom will have 10 possible natural frequency, should these 10 natural frequencies be related to the natural frequencies of the larger model, okay, that's the question. Similarly mode shapes, and we can also ask this question on the frequency response function between 2 coordinates, for example should the frequency response function over a given frequency range of the reduced system serve as acceptable approximations to the corresponding FRF's of the original system, what is the objective of model reduction that we have to specify.

Similarly if you are interested in transient behavior of the system, should transient response to dynamic excitations for the reduced system serve as acceptable approximation to the response of the original system, so the question on how to select sai, therefore gets answered in a broad way that it depends on the situation that context in which you want to reduce the model, so we have to discuss different options that are possible.




Now actually we had talked about model reduction but there can be a problem of model expansion also, how does it come about? Suppose if you consider a structural system that is being studied both experimentally and computationally, if lowercase n is the number of measured degrees of freedom, and capital N is the degrees of freedom in computational model, typically the degrees of freedom in computational model is much greater than the degrees of freedom in an experiment. Now when we are reconciling the prediction from the computational model with what we have measured, then there are 2 options available to us, one is reduce the size of computational model so that only the degrees of freedom which are common to both experimental and computational models are retained, or alternately expand the size of the experimental model so that the degrees of freedom not present in the experimental model are approximately represented, okay so that there is a matching of degrees of freedom between experimental and computational models.


Now some of these issues will be discussing what we will do is, I'd propose to discuss three methods, one is what is known as static condensation, that is also known as Guyan's reduction, and there is a method known as dynamic condensation, and third method is what is known as system equivalent reduction expansion process and it is abbreviated as SEREP, we will be discussing this in the lectures to follow, 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**