Finite Element Method and Computational Structural Dynamics

Prof. Manish Shrikhande
Department of Earthquake Engineering
Indian Institute of Technology, Roorkee

Lecture - 51
The Time Marching- I

Hello friends. So, we now start with a new topic today. We have discussed about the time dimension in the passing and inertia terms when dynamic effects are present. Then, the inertia term will also be there and consistent mass matrix is obtained by assuming acceleration variation within the element to be interpolated in the same way as the displacement interpolations and the mass matrix or inertia matrix that is so derived is known as consistent mass matrix.

Now, consistency this adjective consistent is only with reference to the interpolation nature of interpolation that is being used for interpolating the acceleration field. So, interpolation of acceleration field is consistent with that of the displacement field. Now, after we solve these I mean assemble the global finite element equations, the partial differential equations is eventually converted into an ordinary differential equation in time with some initial conditions.

If the problem is time dependent and if the problem is not time dependent or the time dependence is negligible the time variation is very small or it can be neglected, then the problem is essentially static and the static analysis in simple linear simultaneous equations they serve the purpose the solution.

But when the time effects or dynamic effects are important, then the ordinary differential equation in time needs to be solved. And although in principle, it is possible to develop if we can develop the finite element formulation for spatial coordinates then; obviously, we can also develop the finite element formulation for time coordinates in time dimension. But it makes it simpler and more efficient to distinguish or to separate the variables spatial variables from the time variables and treat the time variable separately.

So, finite element approximation is constructed in the spatial domain x y z domain and the resulting I mean after substituting the finite element approximation, we arrive at the system characteristics or global system matrices, stiffness matrix, inertia matrix and also, the damping matrix suitably constructed.

And. So, resulting second order differential or resulting ordinary differential equation in time needs to be solved using some suitable technique. So, we will start our discussion using the first order differential equation in time which is the basic building block. And second order differential equation acceleration term is essentially a second order differential term and that results in structural dynamics that results in second order differential equation in time.

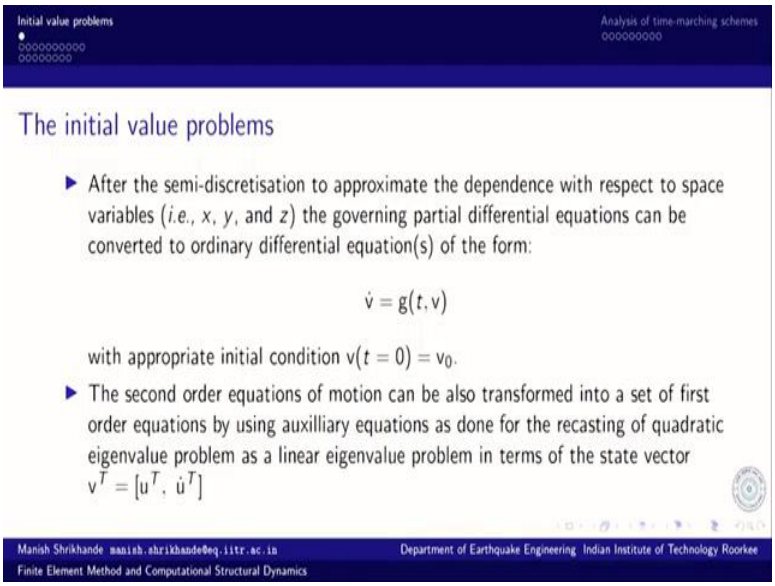
So, this second order differential equation can always be converted into an equivalent first order differential equation and we will see that how it is done. So, this first order differential equation techniques that we discuss can be applied equally well to structural dynamics problem developing I mean resulting in second order differential equations in time.

So, these are called initial value problems and what we deal with finite element formulation, then if we recall the basic finite element approximation, the boundary terms were of crucial importance and those are called boundary value problems. So, that is to distinguish. I mean boundary value problems the problem is bounded from all the sides I mean the domain is bounded with adequate boundary constraints, but the time dimension we have constraints only at the beginning of the time.

There is no constraint at the other end. So, it is open ended domain. So, that is what differentiates boundary value problem from initial value problem. So, now we look at initial value problem or time variation problem and try to find out try to develop numerical tools for solving these time variation problems.

So, after the semi-discretization I mean semi-discretization. I am referring to this separation of variables spatial coordinates are separated from time coordinates and we discretize the space domain spatial domain. Time domain we are not touching. So, that is what we are referring to as semi-discretization. So, discretization using finite element finite element method is done only in the space domain spatial domain.

(Refer Slide Time: 05:42)



So, after the semi-discretization to approximate the dependence with respect to space variables; x y and z the governing partial differential equations can be converted to ordinary differential equations in of variation with respect to time.

And a generic representation is first order derivative d v / d t is given as a function some function g as a function of t the time variable and the unknown variable that is v with appropriate initial condition v at time t is equal to 0. Let us call it v_0.

So, that is the statement of the problem and using this information that we have, the function v that we need to compute has to satisfy this initial condition at time t is equal to 0 that v at time t is equal to 0 should be equal to v_0; some specified value. And the time rate of change of function v is given by f known function g as a function of time and v the current value of the variable v.

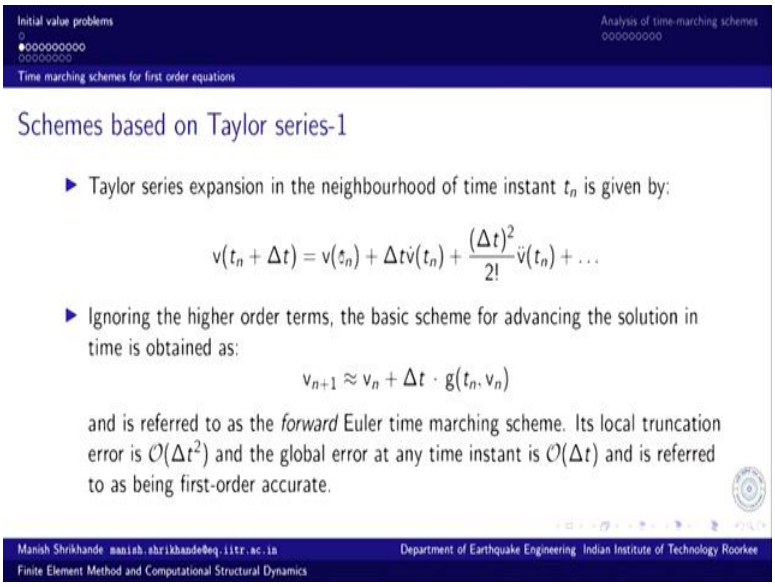
Second order equations of motion that arise in the study of structural dynamics or vibration problems. They can also be transformed into a set of first order equations by using the auxiliary equations as done for we can actually recast the quadratic eigenvalue problem as well in the for an eigen linear eigenvalue problem by defining the state vector.

So, we define a state vector let us say v is defined as a vector of u and $\dot{u}$ so, displacement and velocity. So, the $\dot{v}$ when we talk in terms of rate of change of v then, that will also

include a term involving acceleration and the equations can be arranged so, as to hold the identity and the second order differential equation in terms of displacement u velocity $\dot{u}$ and acceleration $\ddot{u}$ can be converted into state vector v and $\dot{v}$.

Now, returning focus back to this particular equation $\dot{v}$ is equal to g of t and v. So, let us try to derive develop the Taylor series approximation around time t.

(Refer Slide Time: 08:36)



So, function v at some instant t $_n$ plus delta t would be given as shown in the slide. So, this is the Taylor series expansion, in the neighbourhood of the time instant t $_n$.

So, the function v that we need to compute can be expressed as a Taylor series expansion in the neighbourhood of t n the time instant. Now, let us ignore the truncate it after the first order term and ignore the higher order terms for delta t is very small. So, higher order terms can be neglected in comparison with the first-order term.

So, ignoring higher order terms, the basic scheme for advancing the solution in time. So, idea is I mean we start with initial value. So, at time t is equal to 0; this function is known v at t 0 is known and knowing the function, because this function is also known.

So, substituting for v 0, then we also know what is the initial value initial velocity at time t is equal to t 0. So, knowing this $\dot{v}$ at time t is equal to t 0, we can actually find what is going to be the value or we can predict what should be the value of v at next time step by incrementing the contribution coming from the derivative.

So, that is what we obtained ignoring the higher order terms. The basic scheme for advancing the solution in time is obtained as v_{n+1} is equal to the next time step the value of the function at next time step is approximately equal to v_n the value of the function at current time step t_n plus Δt times g of t_n v_n, because this function is known as we know this function as a function of t and v for $\dot{v}$.

So, we substitute this for the value of t and v at t_n and this is the first order approximation for unknown value at next time instant. And then, we can go back and this becomes a recursive relationship. So, for computing n + 2 we substitute here, instead of v n we write v_{n+1} and here, we substitute t_{n+1} and v_{n+1} and so on it goes.

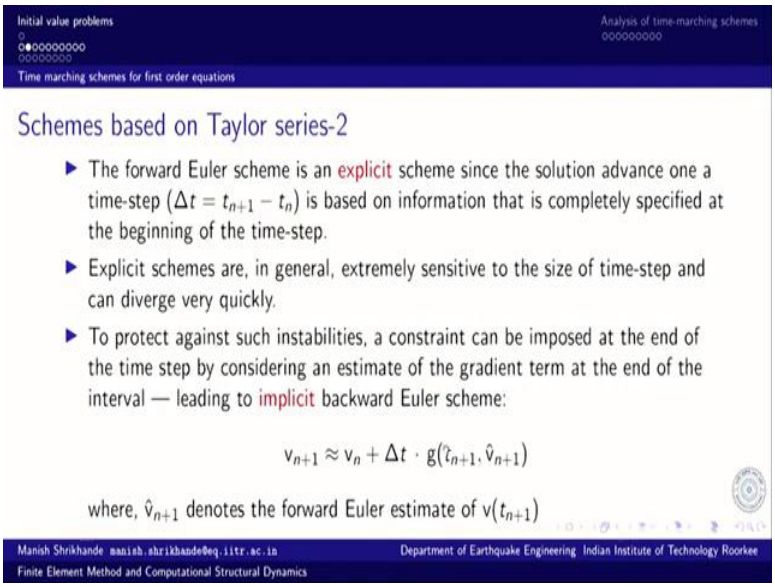
So, this is referred to as forward Euler time marching scheme. I mean we are marching ahead in time which one step at a time, we are projecting the solution in the next time step. And knowing the value I mean knowing the function of the derivative function or derivative and knowing the time instant and the function value at the previous time instant, we can estimate what is the derivative at the time step and we can add increment, because of this gradient function to the function value over the time step.

So, this is referred to as forward Euler time marching scheme and its local truncation error I mean we truncated it at the higher order term second order term. So, local truncation error is of the order of delta t squared and global error at any time instant, because it is going to be added up accumulated. So, global error will be little less accurate.

So, it will be of the order of Δt and it is referred to as being first-order accurate scheme. And therefore, first order accuracy is not a very good accuracy and this has although, this is very simple to implement. I mean very trivial to code on a computer actually and; obviously, nothing comes for free. So, if something numerical technique is such a simple algorithm to code then; obviously, there has to be some catch. And the catch is this is this technique is numerically unstable. The problem is there is no constraint on the next time instant on which it is being the solution is being predicted. All information is based on the previous time instant and there is no protection against what is happening within the interval that we are projecting the solution ahead in time over the interval.

So, whatever is the mechanics happening, we have absolutely we are not taking into account anything about that interval. We are projecting solution in the next instant of time based on entirely on the information available at the previous instant; that is all.

(Refer Slide Time: 14:18)



So, forward Euler scheme and this is known as explicit scheme. Since, the solution advance one at step at a time is based on information that is completely specified at the beginning of the time step. So, there is no uncertainty as far as the information is required to for the prediction to next time step.

So, this is called explicit scheme and as we have seen that is a general character of explicit schemes, they are in general very easy to compute and they are very efficient in the sense that they do not require too much of computation. But they do come at a the offside being they do come at a penalty incur a penalty in terms of tendency to be unstable at times.

So, explicit schemes are in general extremely sensitive to the size of time step and can diverge very quickly as I said they can they have a tendency for instability. So, to protect against such instabilities, a constraint can be imposed at the end of the time step by considering an estimate of the gradient term at the end of the interval.

So, this gradient term; instead of using this gradient term at the beginning of the interval that is at t n, we make use of this gradient term I mean invoke this gradient term at the

end of the interval. So, $t_n + 1$. Now; obviously, this gradient term g requires information about v as well. So, this $\hat{v}$ that we use as a value at n + 1.

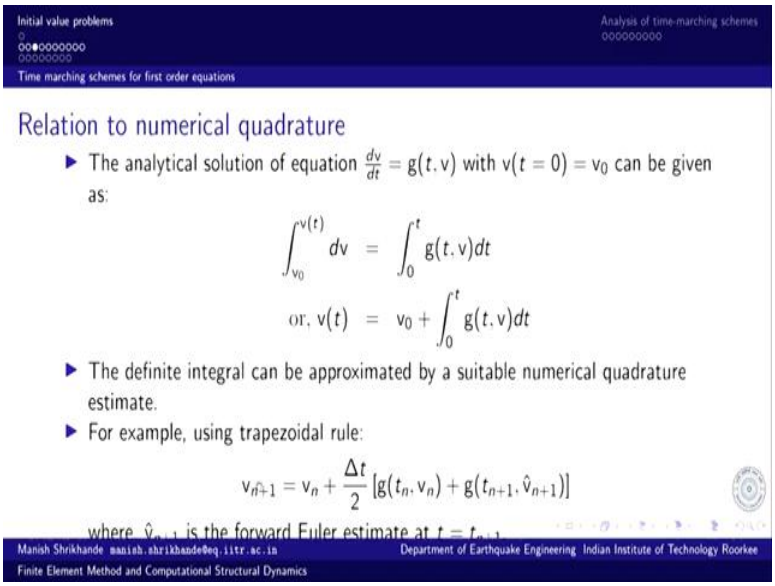
That can be taken as a Euler forward Euler estimate of v_n. So, in order to compute this, we use Euler estimate forward Euler estimate. And then, use gradient that estimate to compute the gradient at the end of the interval and then use it for computing the next value of v at next time instant and this is known as the backward Euler algorithm implicit backward Euler scheme.

And this is implicit, because we are using information at the end of the time step at which the solution is being computed. So, we are computing solution at n + 1th times step and we are using information at n + 1^{th} time-step in addition to the information at the previous time step. So, this is an implicit scheme it is based on the information at the current time step at which the solution is being predicted. So, this; obviously, has a I mean the constraint is imposed at the end of the time step.

So, it prevents it from diverging and gradient information is captured at the end of the interval and that has a much important much better performance over the forward Euler scheme and still the algorithm I mean little bit of computational effort is increased. We need to estimate this time marching twice; first using forward Euler estimate and then use that estimate in the gradient function and then, use that gradient estimate gradient function at the end of the interval to compute the next estimate of the forward Euler scheme; so, backward Euler.

And that results in this particular constraint at the end of the time interval results in considerable improvement in the stability characteristics of the scheme. So, now, we come to more generic way. I mean this is of course, coming straight forward from Taylor series expansion or even finite difference approximation, we could have formulated this. So, more generic formulation is based on.

(Refer Slide Time: 18:45)



The analytical solution d v / d t is equal to g (t) and v with v t is equal to 0 as v_0. It can be given analytically we can generate it as v_0 that is the constant of integration plus 0 to t integral from 0 to t of this gradient function. Now, the entire set of numerical schemes can be developed by using this how to evaluate this integral definite integral. Now, this definite integral I mean we have already seen it before all definite integrals in on a digital computer. They are evaluated using quadrature rules.

So, we have seen several quadrature rules that have been earlier I mean rectangle rule trapezoid rule Gauss quadrature Newton code Simpson rule and so on. So, this definite integral can be evaluated using several different quadrature schemes. And each of those has its own convergence or accuracy properties I mean error term and so on and that will have effect on the quality of computation.

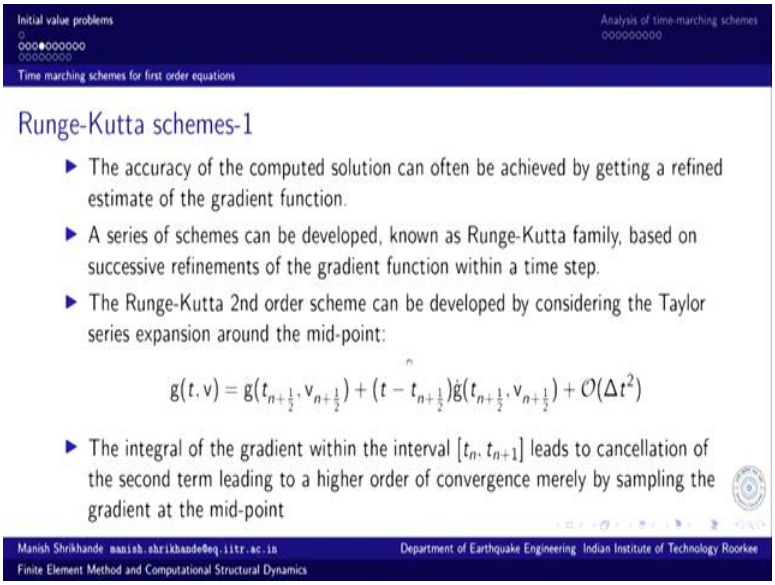
So, definite integral can be approximated by a suitable numerical quadrature estimate and if we assume trapezoidal rule, then the trapezoidal rule can be invoked and the function value at next time instant v_{n+1} can be given as $v_n + \Delta t / 2$ and average of the function values gradient values at the two-time instants.

So, g at t_n v_n and g at $t_n + 1$ and $\hat{v}_n$ on where $\hat{v}_n$ is the forward Euler estimate at t is equal to t_n plus 1. So, once we have this forward Euler estimate, then we can invoke trapezoidal rule and this can be invoked. So, this is an improvement I mean if you look at it, this is similar to backward Euler except that instead of having only function of I mean

the gradient evaluated at end of the time interval, we are looking at average of the gradient over the at the beginning and at the end of the time interval and evaluate it the contribution to get the function value at the next time instant.

So, next in line comes very popular scheme popular family of schemes which are proposed by Runge and Kutta.

(Refer Slide Time: 21:45)



So, there are of different orders and essentially the whole point is concentrated the entire scheme is concentrated on how do we estimate this gradient function. This quadrature I mean the whole crux of the problem is if v_0 is known how accurately can I compute this integral or this integral. And for that we need to know, because the function this gradient actually depends on the function v itself and v is not known; v is known only at 0 and in between this duration this function v is not known.

So, the entire effort now I mean Runge-Kutta the algorithms; they are based on how to approximate how to get a better degree of approximation of this gradient function within this interval over which it is being integrated. So, 0 to I mean t_n to $t_n + 1$ during this interval how do we approximate this gradient function.

So, accuracy of the computed solution can often be achieved by getting a refined estimate of the gradient function. Series of schemes can be developed known as Runge-Kutta family based on successive refinements of the gradient function within a time step.

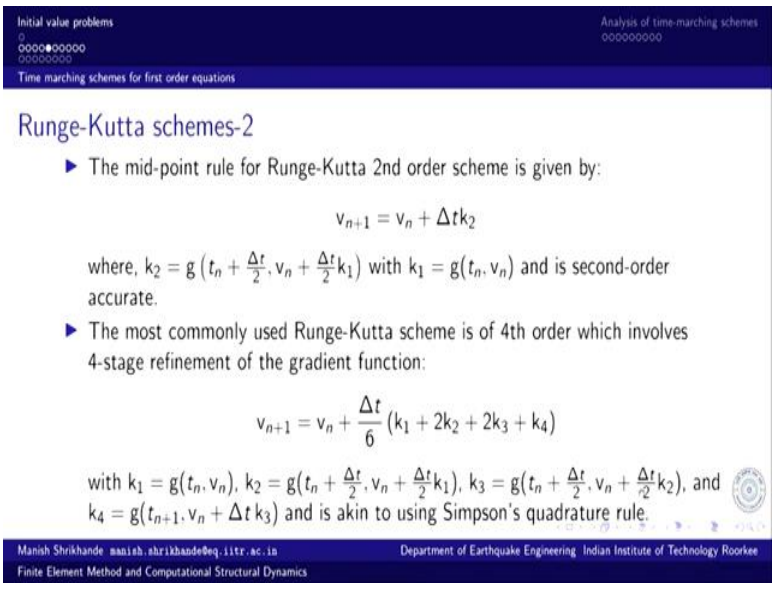
So, from t_n to $t_n + 1$ within this time step, how the gradient function varies. And 2nd order scheme Runge-Kutta 2nd order scheme can be developed considering the Taylor series expansion Taylor series of the gradient function around the midpoint of the interval. So, let us denote the midpoint as n plus half. So, gradient function at t_n plus half will of course, depend on the function value at n plus half.

And then, it will be of course, interval $t - t_n + (1/2)$ and then gradient function the gradient of the gradient function g dot and higher order terms I mean delta t square terms. So, integral of gradient within the interval t_n to $t_n + 1$ leads to cancellation of the second term; this term cancels out, because of this term being positive at one half and negative for the other half.

And therefore, this leads to a higher order of convergence merely by sampling the gradient at the midpoint; just, because we are sampling the gradient value at the midpoint, allows us for cancellation of this term I mean first order term and the convergence accuracy is increased by one order.

So, that is a very clever construction and this midpoint evaluation can of course, be evaluated by using several approximations.

(Refer Slide Time: 25:06)



And that is done by using successive gradient values. So, k_1 let us say k_1 is the grade value of the gradient at beginning of the time instant t_n and v_n. And at the midpoint, the

gradient k_2 is defined as g at $t_n + \Delta t / 2$ and $v_n \Delta t / 2$ times k_1. So, this is essentially forward Euler estimate for predicting the value of v at midpoint.

So, essentially k_2 is the value of the gradient considering the forward Euler estimate for value of the variable v at midpoint using the gradient function at the beginning of the time instant. And then, using this gradient value, we can estimate the second the prediction; that is $v_n + 1$ is equal to $v_n + \Delta t$ times k_2.

So, that is the midpoint gradient and that is by virtue of that sampling at midpoint it gives us higher order convergence. The most commonly used and we go to higher order term. And most commonly used Runge-Kutta scheme is of 4th order its 4th order accurate and it involves four stages of refinement of the gradient function.

So, we keep on repeatedly refining the gradient function. So, k 1 is of course, the at the beginning of the interval. Using k 1, we find out what is k 2, then using k 2, we find out what is k 3. So, just separating the I mean again forward Euler estimate, we can we keep on changing the value of the gradient term by using the previously used improved estimate of the gradient term.

So, here you can see these gradient terms k_1 is the beginning of the interval, k_2 is based on the gradient term at the beginning of the interval. So, this is forward Euler estimate. And k_3 again based on the previously refined estimate of the gradient k_2 and evaluated at the midpoint. And then finally, k_4 is evaluated at the end of the time step. So, $t_n + 1$.

And we make use of this previous refined estimate of gradient function to find out what is the again forward Euler scheme to find out what should be the value of the function at the estimate of the function value at the end of the time step. And then, use these in the Simpson rule like formulation.

So, next time instant value at the next time instant $v_n + 1$ is given as v_n previous time instant + $\Delta t / 6$ multiplied by now, weighted some of the gradient values. So, $k_1 + 2 k_2 + 2 k_3 + k_4$. So, that is a weighted some of gradient values and that is averaged over it. All the sum of weights is 6 and this is a considered to be a much refined estimate of the gradient function and the value that we get is very accurate.

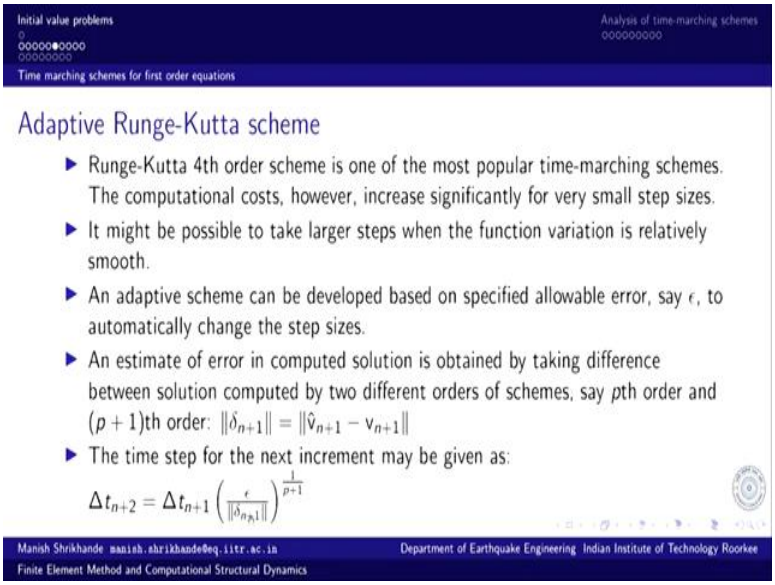
And it is indeed the most efficient and most sort after numerical technique for solving non-linear problems particularly very sensitive problems involving non-linear dynamics and chaotic systems; Runge-Kutta 4th order scheme is the default go to algorithm to solve those problems.

Of course, we can there are several possibilities here. We can dynamically change the time steps by keeping track of the error. So, that is called Runge-Kutta 4 5 r k 4 5. If you look at several MATLAB or octave implementation, there will be a function ode 4 5 that actually is solution of this ordinary differential equation in time initial value problem, using Runge-Kutta 4th order method and 5th order method and that 4 5 is actually this 4th order and 5th order Runge-Kutta estimate.

So, 5th order is just to estimate the error compare, because that is an higher order method. So, that is difference between 4th order estimate and 5th order estimate is an indication of the error in computation. So, wherever error increases above the acceptable limit, time steps can be reduced and when the error goes below acceptable limit, time step can be increased.

And that is how adaptive time stepping can be achieved, in case of Runge-Kutta schemes.

(Refer Slide Time: 30:37)



So, adaptive Runge-Kutta scheme. So, 4th order scheme is one of the most popular time-marching schemes. The computational costs, however, increase significantly for very small step sizes. And therefore, we would like to take as large steps as possible whenever the function variation is relatively smooth.

So, how do we keep track? So, an adaptive scheme can be developed based on specified allowable error say epsilon to automatically change the step sizes. The estimate of error in computed solution is obtained by taking difference between the solution computed by two different orders of schemes. So, let us say p^{th} order and $p + 1^{th}$ order and that is what I referred to as o d e for 5 4th order scheme and 5th order scheme.

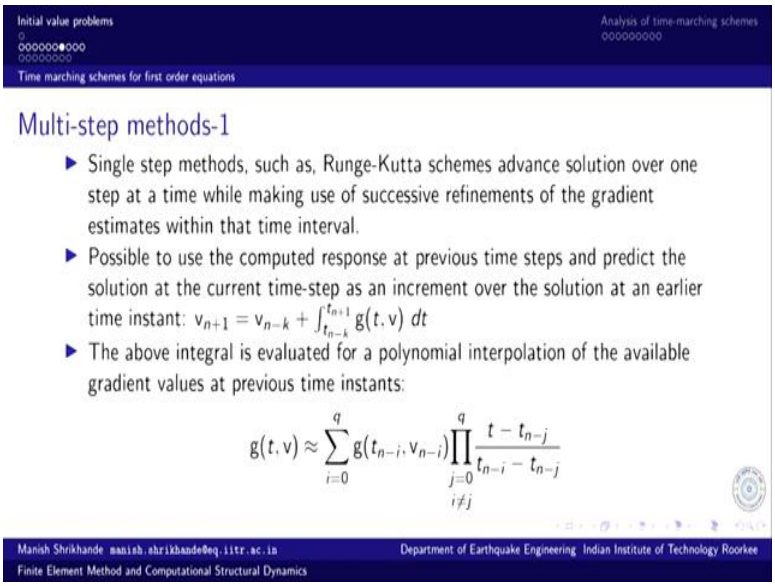
So, the computation of difference or the size of the difference between a higher order result from the lower order result, that is an estimate of the error in computation. And if this error is large, the computation I mean based on this error computed error, the time step for the next increment can be automatically selected by taking it as a fraction of or scaling up or down of the time step for the previous calculation.

So, $\Delta t_n + 1$ can be scaled by this ratio. Epsilon is the tolerance level and $\Delta n + 1$ is the error in computation over the last increment. And if delta is smaller than epsilon, then this scheme will automatically allow us to take larger interval larger step size. If error is larger than epsilon, then it will try it will give us smaller step size appropriately.

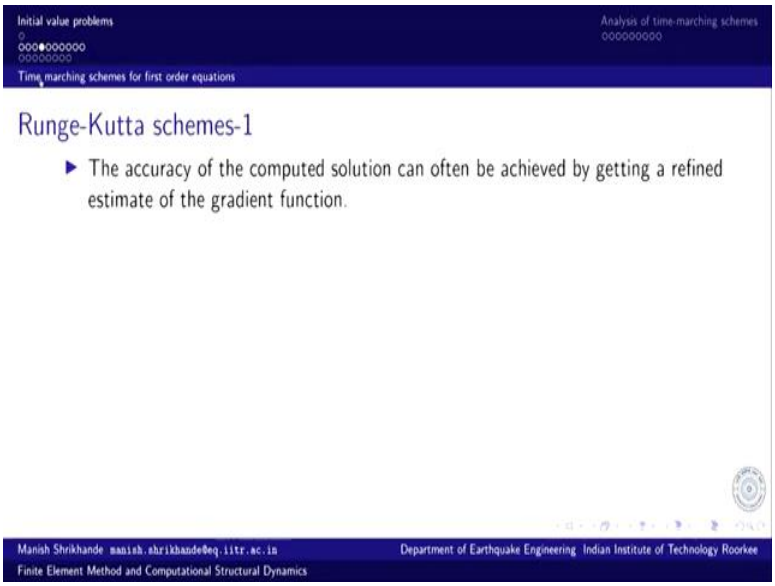
And this adaptive time scheme generally gives very good results and result I mean very highly dynamical system highly non-linear dynamical systems can be analyzed very accurately by using Runge-Kutta schemes. Now, these are all. So, far we have dealt with these are all single step methods; Euler scheme, Runge-Kutta scheme, because they are projecting making use.

I mean all these methods make use of the values or the information available at the previous time instant and knowing the current time instant, we predict what is going to happen at the next time instant and that is all. We these methods do not really bother or do not really require information about anything beyond one step back right. So, for to compute the solution at $n + 1$, we only need information that is available at n^{th} time instant.

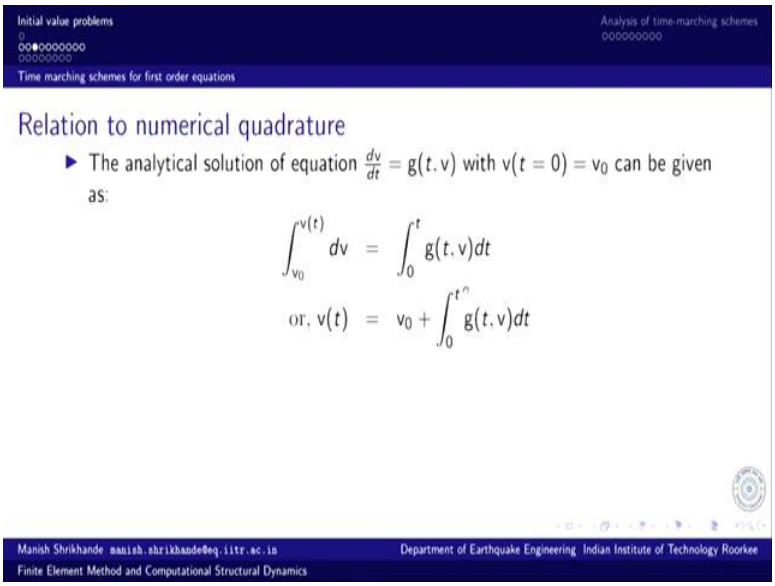
(Refer Slide Time: 34:15)



(Refer Slide Time: 34:17)



(Refer Slide Time: 34:18)



We do not really need information about n - 1th time instant and n - 2th time instant, but that there is a question. I mean if the whole point is to obtain an approximation of this gradient function, what is wrong if I have the information and I can generate an interpolation and then, extrapolate over one step to get a estimate of gradient function g and then develop analytical solution.

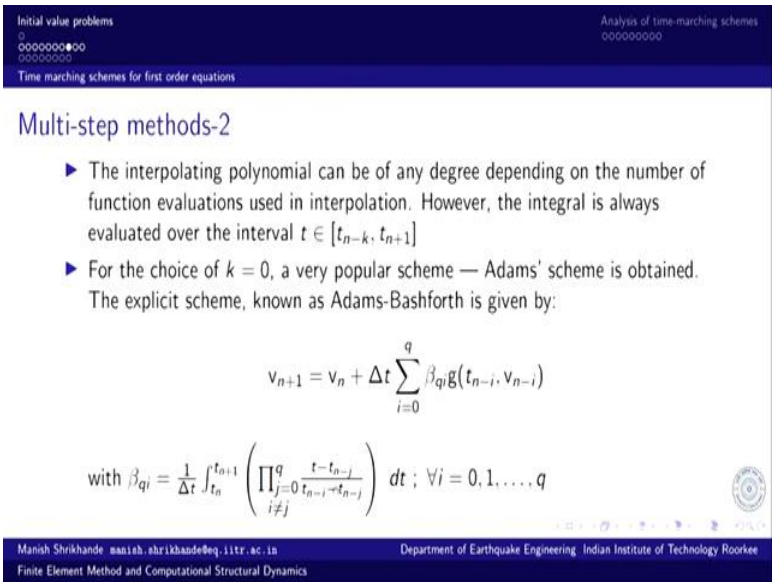
I have interpolated function and that can be integrated analytically and develop the solution. And that is what leads to multi-step methods, where we make use of information over the several previous time instants and try to construct suitable approximation over the gradient function.

So, possible to use computed response at previous time instants and predict solution at the current time step as an increment over the solution at an earlier time instant that is $v_{n+1} = v_{n-1} +$ the integral v_{n-k}; k steps back. And then, integrate this gradient function from between time instant n - k to n + 1. So, the entire function can be approximated.

So, now, we have this gradient function, because we know the function values at various previous time instant. So, gradient functions information is also available and that allows us to construct a reasonably good approximation for the gradient function over this interval which goes beyond the current instant and unknown territory of next time instant.

So, above integral is evaluated. For a polynomial interpolation of the available gradient values at previous time instants. So, we can have this as a just as in Lagrangian interpolation model and the gradient function can be evaluated as a interpolation of the previously computed gradient values at previous time instants. And once we have this gradient function available analytical form, we can evaluate it and suitable coefficients can be generated.

(Refer Slide Time: 36:54)

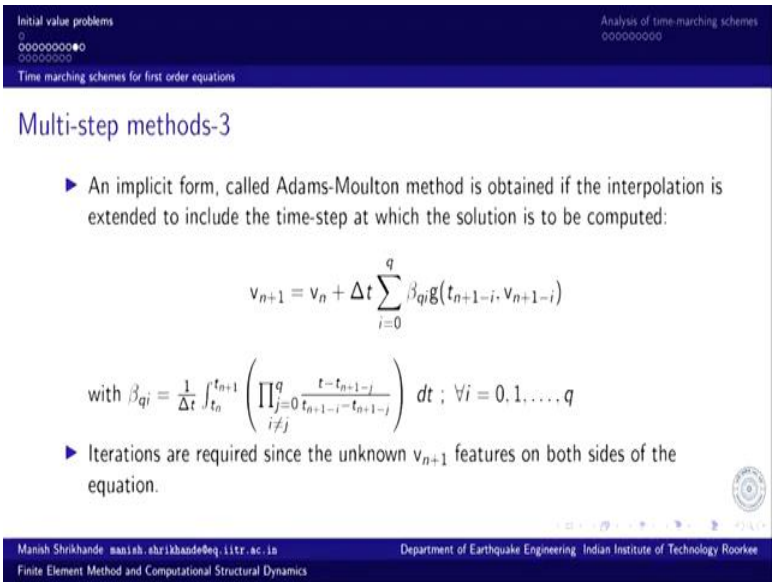


So, interpolating polynomial can be of any degree depending on the number of function evaluations used in the interpolation. However, the integral is always evaluated. So, whatever may be the degree of polynomial. I mean that depends on how much steps back we are going. So, whatever may be the degree of polynomial interpolation for the approximation of gradient function, the integral is always evaluated over the interval n - k to n+ 1.

And if we choose k = 0 that is no backward I mean just the previous step and we get what is known as Adams scheme and the explicit I mean that is an explicit scheme and known as Adams-Bashforth. And that is based on just the previous instant value and coefficients can be evaluated by using this interpolation model.

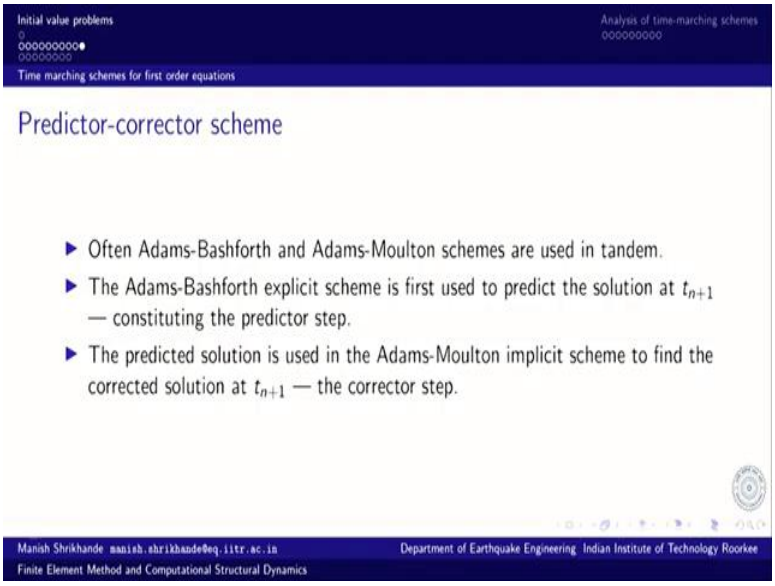
So, several time instants in between the time interval can be developed and approximations can be generated. And this βq_i can be evaluated as an integral of these Lagrangian interpolation models.

(Refer Slide Time: 38:36)



An implicit form called Adams-Moulton is obtained if the interpolation is extended to include the time step at which the solution is to be computed. So, we can have the interpolation that I mean gradient function is also interpolated at the next time instant. And in this particular ray's the iterations may be required, because the solution v $_{n}$ + 1 is not known on and it figures on both sides of the equation. So, iterative scheme is required to compute the value of v $_{n}$.

(Refer Slide Time: 39:15)



And often Adams-Bashforth and Adams-Moulton scheme they are used in tandem; one is explicit. So, that is used as a predictor and Adams-Moulton is used as a corrector of this predicted solution. So, constituting the predictor step, Adams-Bashforth and Adom Moulton is an implicit scheme to find the corrected solution the corrector step.

So, we stop here and we will discuss the direct solution of 2nd order differential equation in time in our next lecture.

Thank you.