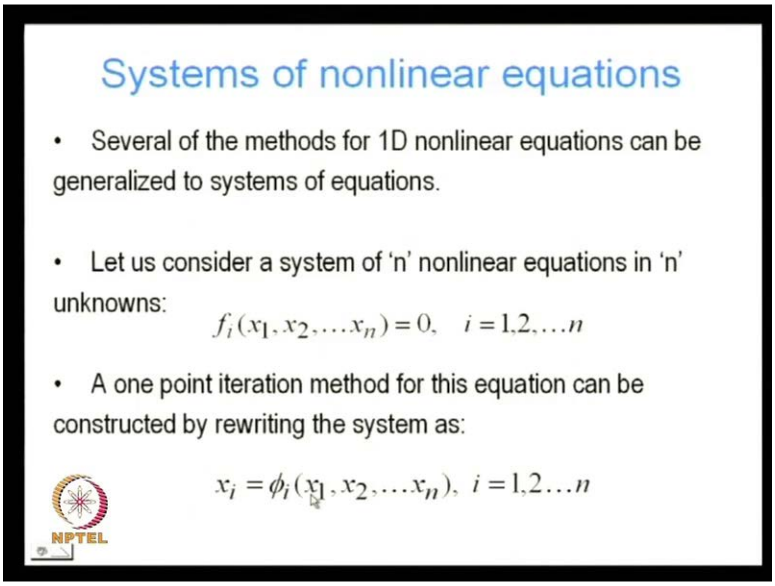
Numerical Methods in Civil Engineering

Prof. Arghya Deb
Department of Civil Engineering
Indian institute of Technology, Kharagpur

Lecture - 14
Solving Multi Dimensional Nonlinear Equations

(Refer Slide Time: 00:29)



Systems of nonlinear equations

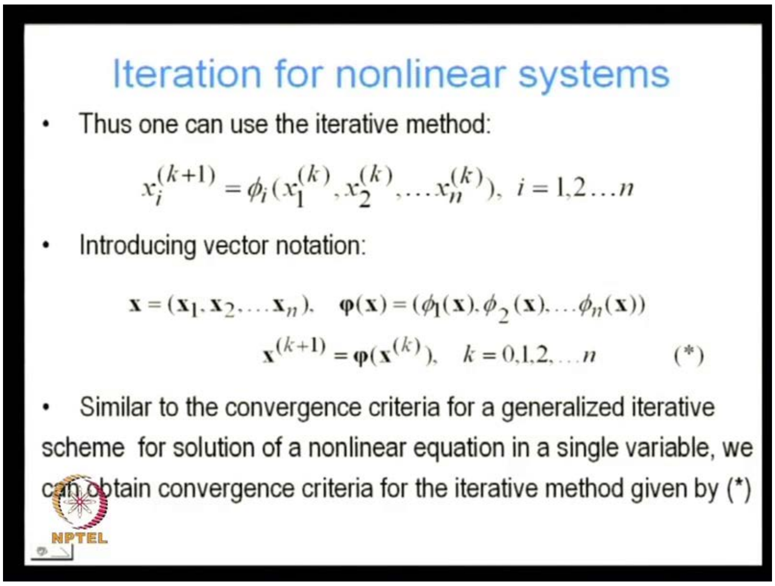
- Several of the methods for 1D nonlinear equations can be generalized to systems of equations.
- Let us consider a system of 'n' nonlinear equations in 'n' unknowns: $f_i(x_1, x_2, \ldots x_n) = 0, \quad i = 1,2,\ldots n$
- A one point iteration method for this equation can be constructed by rewriting the system as:

$$x_i = \phi_i(x_1, x_2, \ldots x_n), \; i = 1,2\ldots n$$

We will talk about solving multi-dimensional non-linear equations. In the last lecture, we focused in the last two lectures, we focused on 1D non-linear equation, and we looked at methods and solutions, various algorithms for solving those equations. We looked at the convergence properties when those algorithms are going to convert, and we also looked at the order of convergence of these algorithms the rate at which we are going to convert. So, this time we are going to focus on multi-dimensional non-linear equations. And we will see that many of the techniques, many of the algorithms we looked at for 1D non-linear equations, they carry over multi dimensional non-linear equations in a more general form.

So, let us consider a system of 'n' non-linear equations in n unknowns. We have 'n' equations. Sample equation is like this. If i is equal to f 1, f 2, f 3 and so on up to f n, it will be n equations and each equation is in 'n' unknowns. So, then we have a system of n equations in 'n' unknowns f 1, f 2, f 3 up to f n and variables are x1 through x n. One point iteration method for this equation can be constructed by rewriting the system like this we saw for the 1D non-linear equation. How we can rewrite is f x equal to 0 to x is equal to f x. If we can write that, we can construct an iterative scheme. Similarly, in multi-dimensions we construct a series of equations, ‘n’ equations. In fact, x 1 is equal to phi 1 of x 1 x 2 through x n, x 2 is equal to phi 2 of x 1 through x n and so on and so forth. So, again we frame an equation. We are just rewriting them as x i is equal to phi i of x 1 through x n.

(Refer Slide Time: 02:32)



Iteration for nonlinear systems

- Thus one can use the iterative method:

$$x_i^{(k+1)} = \phi_i(x_1^{(k)}, x_2^{(k)}, \dots x_n^{(k)}),\ i = 1,2\dots n$$

- Introducing vector notation:

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_n),\quad \boldsymbol{\varphi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots \phi_n(\mathbf{x}))$$

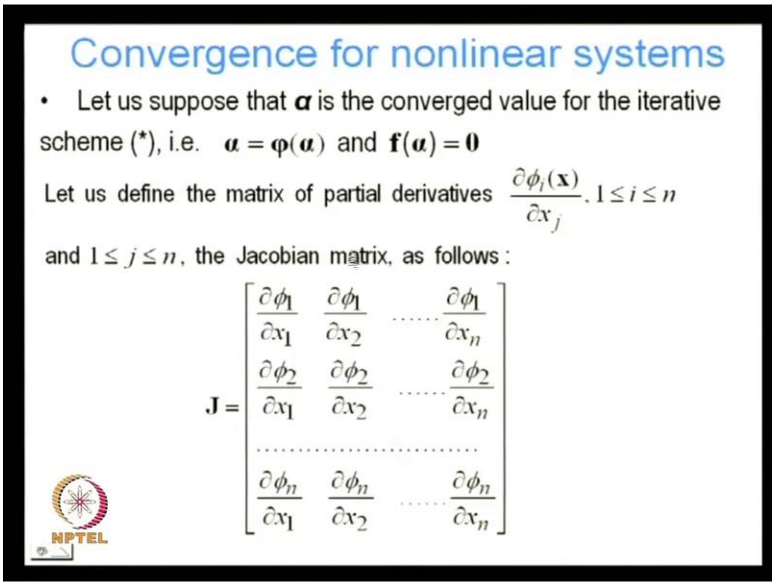
$$\mathbf{x}^{(k+1)} = \boldsymbol{\varphi}(\mathbf{x}^{(k)}),\quad k = 0,1,2,\dots n \qquad (*)$$

- Similar to the convergence criteria for a generalized iterative scheme for solution of a nonlinear equation in a single variable, we can obtain convergence criteria for the iterative method given by (*)

Once we do that, we can use the iterative method, but now instead of having an iterative method for one equation, we have an iterative method for n equations and n unknowns. So, that iterative method will be like this, x i at k plus 1. So, by new value of x is equal to phi i, that function evaluated with my old values of x and I keep on iterating until it converges. So, introduce vector notation, we have this vector x which basically means x 1, x 2 through x n. This should not be in bold because x 1, x 2, x n are scalars, right. So, vector x is a vector while these are scalars while phi of x that is a vector and that comprises these scalars phi 1, which is a scalar valued vector function. It is a function of a vector x, but its value is a scalar. So, it is a scalar valued function of a vector, right.

So, phi 1 of x phi 2 of x phi n of x, these are phi's and then we have in vector notation I can write the iterative scheme as x k plus 1 is equal to phi of x of k, right. So, we have n equations like this, x 1, x 2 and so on and so forth and this is going to hold for k equal to 1, 0 to n. So, similar to the convergence criteria for the enveloped iterative algorithm, we can construct a convergence criteria for multi-dimensional non-linear equations also.

(Refer Slide Time: 04:22)



Convergence for nonlinear systems

- Let us suppose that $\boldsymbol{\alpha}$ is the converged value for the iterative scheme (*), i.e. $\boldsymbol{\alpha} = \boldsymbol{\varphi}(\boldsymbol{\alpha})$ and $\mathbf{f}(\boldsymbol{\alpha}) = \mathbf{0}$

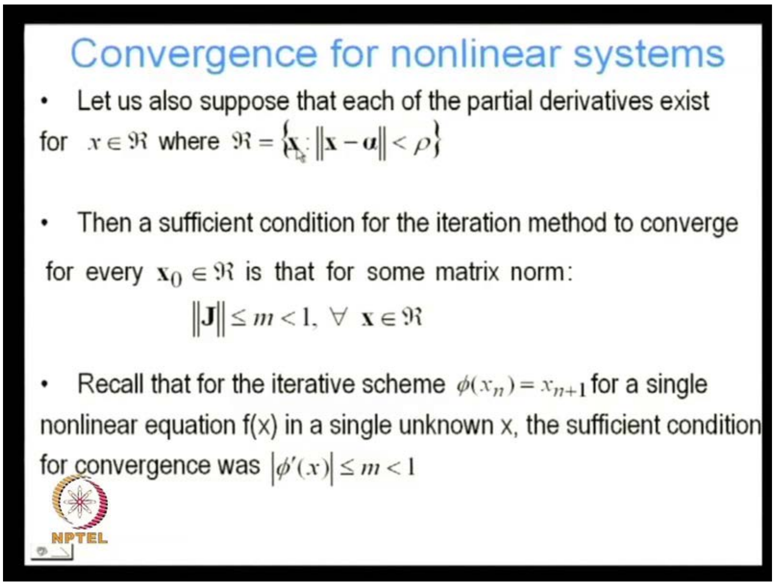
Let us define the matrix of partial derivatives $\dfrac{\partial \phi_i(\mathbf{x})}{\partial x_j}$, $1 \le i \le n$ and $1 \le j \le n$, the Jacobian matrix, as follows :

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \phi_1}{\partial x_1} & \dfrac{\partial \phi_1}{\partial x_2} & \cdots\cdots & \dfrac{\partial \phi_1}{\partial x_n} \\ \dfrac{\partial \phi_2}{\partial x_1} & \dfrac{\partial \phi_2}{\partial x_2} & \cdots\cdots & \dfrac{\partial \phi_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial \phi_n}{\partial x_1} & \dfrac{\partial \phi_n}{\partial x_2} & \cdots\cdots & \dfrac{\partial \phi_n}{\partial x_n} \end{bmatrix}$$

So, let us suppose that alpha is the converged value for the iterative scheme that is alpha is equal to phi of alpha. So, once it converges, phi of alpha is equal to alpha. So, that is my root, right and f of alpha is equal to 0 because that is my root 2, right. So, let us define metrics. When we look at 1D neutron Raphson for 1D non-linear equations, we looked at the derivative of phi with respect to x 1 variable, but now we have multiple phi and we have multiple x, right. So, we are going to have a metrics of partial derivatives.

What will be the terms in the matrix like del phi, del phi 1, del x 1, del phi 1, del x 2 and so on and so forth. So, the first row will be the derivative, partial derivatives of phi 1 with respect to x 1 through x n. The second row will be partial derivatives of phi 2 with respect to x 1 through x n and so on and so forth. So, it will be an n by n matrix if I have n phi s and n x, right and this matrix typically is known as Jacobian matrix. It is a Jacobian matrix.

(Refer Slide Time: 05:40)



Convergence for nonlinear systems

- Let us also suppose that each of the partial derivatives exist for $x \in \Re$ where $\Re = \{\mathbf{x} : \|\mathbf{x} - \boldsymbol{\alpha}\| < \rho\}$
- Then a sufficient condition for the iteration method to converge for every $\mathbf{x}_0 \in \Re$ is that for some matrix norm:

$$\|\mathbf{J}\| \le m < 1, \ \forall \ \mathbf{x} \in \Re$$

- Recall that for the iterative scheme $\phi(x_n) = x_{n+1}$ for a single nonlinear equation f(x) in a single unknown x, the sufficient condition for convergence was $|\phi'(x)| \le m < 1$

So, let us suppose that each of the partial derivatives. Now, let us look at the convergence of the non-linear systems. We looked at convergence of linear systems and we found during the, sorry convergence of non-linear equations of one variable and there we found that if in a neighborhood close to the root we impose certain conditions on my derivatives, the derivatives are bounded, right.

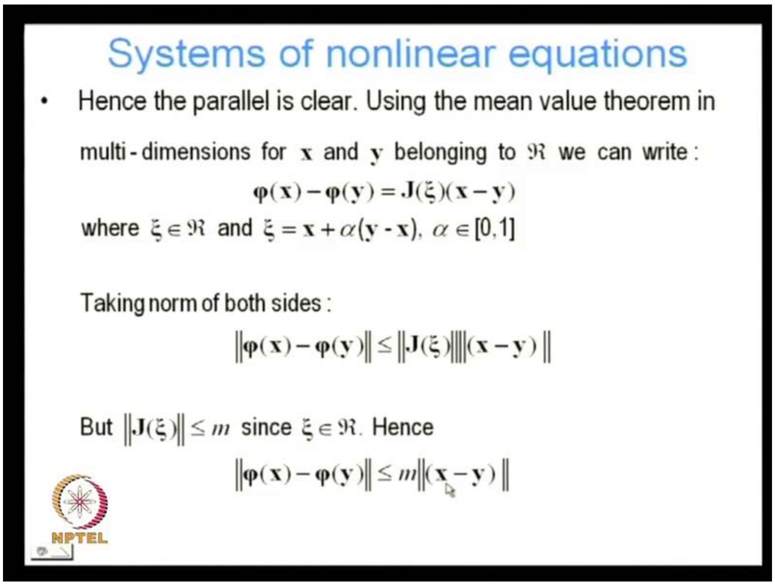
We said the derivatives are bounded in the neighborhood of the root and in that case, we can show that if we start with an iterate in that neighborhood, then we are going to converge to the root, right and we are going to always stay within that neighborhood. We are not going to leave that neighborhood and we are going to converge to the root to the solution, right.

So, each of the partial derivatives in my Jacobian, they exist in a neighborhood of the root, right and that neighborhood I am calling r, right. How am I defining that neighborhood? That neighborhood is r, such that x, any of my candidates solution is within a sphere, within a hyper sphere. Actually it is not a sphere because this is n dimensional space within a hyper sphere whose center is at the root alpha, n dimensional vector with center at n dimensional vector alpha and then I have a hyper sphere center at alpha with radius which is rho, right. I am saying that I am looking at all those x which lie in that r in that neighborhood of the root, right and I am imposing the condition that each of the partial derivatives exist all those partial derivatives del phi i del x J, they exist in that neighborhood. Then the sufficient condition for the iteration method to converge for every x 0 belonging to r is that.

For some matrix norm, the norm of the Jacobian matrix must be less than m where m is the number which is less than 1, right. So, the criterion is that Jacobian matrix and then I evaluate the norm of the Jacobian matrix. We looked at several definitions of matrix norms, right. We looked at several definitions of matrix norm earlier in the course. Suppose we take one of those norms, rinses the infinite norm, right. May be the l 2 norm, right, any of those norm so long in some consistent norm says nothing about the type of the norm, right. If in any norm this is less than m which is less than 1, for all the x belonging to r.

So, this norm of the Jacobian matrix is less than 1 is strictly less than 1, then we are going to get convergence that is the sufficient condition for convergence. Let us recall that for the iterative scheme that one-dimensional non-linear equation. The single unknown, the sufficient condition for convergence was norm of phi prime of x is less than m less than 1 is just generalization, right. So, instead of mod here, I now have a matrix norm, right. I have a matrix norm and this matrix norm must be less than 1.

(Refer Slide Time: 09:26)



Systems of nonlinear equations

- Hence the parallel is clear. Using the mean value theorem in multi-dimensions for $\mathbf{x}$ and $\mathbf{y}$ belonging to $\Re$ we can write:

$$\boldsymbol{\varphi}(\mathbf{x}) - \boldsymbol{\varphi}(\mathbf{y}) = \mathbf{J}(\boldsymbol{\xi})(\mathbf{x} - \mathbf{y})$$

where $\boldsymbol{\xi} \in \Re$ and $\boldsymbol{\xi} = \mathbf{x} + \alpha(\mathbf{y} - \mathbf{x}),\ \alpha \in [0,1]$

Taking norm of both sides:

$$\left\|\boldsymbol{\varphi}(\mathbf{x}) - \boldsymbol{\varphi}(\mathbf{y})\right\| \leq \left\|\mathbf{J}(\boldsymbol{\xi})\right\|\left\|(\mathbf{x} - \mathbf{y})\right\|$$

But $\left\|\mathbf{J}(\boldsymbol{\xi})\right\| \leq m$ since $\boldsymbol{\xi} \in \Re$. Hence

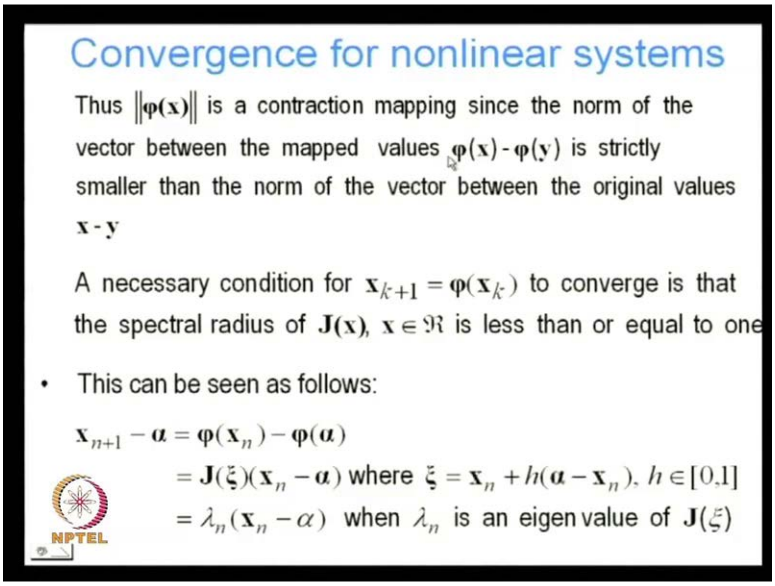
$$\left\|\boldsymbol{\varphi}(\mathbf{x}) - \boldsymbol{\varphi}(\mathbf{y})\right\| \leq m\left\|(\mathbf{x} - \mathbf{y})\right\|$$

Hence, the parallel is clear using the mean value theorem in multi-dimensions, where x and y belonging to r. So, we looked at the mean value theorem in 1D earlier around. So, now, we are looking at the mean value theorem in multiple dimensions, right and we say using that mean value theorem and for any x and y which belong to my little neighborhood r centre around alpha, and with radius row I can write phi x minus phi y is equal to J of psi times x minus y. Recall the parallel. What was the parallel? It was phi of x small x, where x is a scalar minus phi of y is equal to del phi del x evaluated at psi times x minus y, right. So, that was my 1D mean value theorem. So, this is my multi-dimensional mean value theorem, where this is now a vector that is a vector valued function of a vector, that is a matrix and that is again a vector, right.

What is psi? Psi must belong to that space because of the mean value theorem. Psi must lie between x and y, right. So, psi must belong since both x and y belong to r and since psi must be between x and y, psi also must belong to r, right. How do we define psi? Psi is equal to x plus alpha times y minus x, where alpha belongs to 0 1. So, starting I am drawing. Think of it in 3D if it has to understand, right. I am drawing a vector from x to y, right and my psi, my vector value, my vector psi lies somewhere on that on that line, right. So, it is x plus alpha some scalar times, the vector between x and y, right.

So, that is my psi, right. So, taking the norm on both sides again, we can take any norm, right. For instance, let us suppose we take the infinite norm on both sides. So, norm of phi x minus phi of y is going to be lesser than or equal to norm of Jacobian evaluated at psi times norm of x minus y. We have seen that before many times, but we have seen the condition for convergence is norm of J of psi is less than m, right and we have seen that this is true. So, since this is lesser than since psi belongs to r, this must be lesser than or equal to m since norm of J of psi is less than m, right.

(Refer Slide Time: 12:29)



Convergence for nonlinear systems

Thus $\|\boldsymbol{\varphi}(\mathbf{x})\|$ is a contraction mapping since the norm of the vector between the mapped values $\boldsymbol{\varphi}(\mathbf{x}) - \boldsymbol{\varphi}(\mathbf{y})$ is strictly smaller than the norm of the vector between the original values $\mathbf{x} - \mathbf{y}$

A necessary condition for $\mathbf{x}_{k+1} = \boldsymbol{\varphi}(\mathbf{x}_k)$ to converge is that the spectral radius of $\mathbf{J}(\mathbf{x})$, $\mathbf{x} \in \Re$ is less than or equal to one

- This can be seen as follows:

$$\begin{aligned} \mathbf{x}_{n+1} - \boldsymbol{\alpha} &= \boldsymbol{\varphi}(\mathbf{x}_n) - \boldsymbol{\varphi}(\boldsymbol{\alpha}) \\ &= \mathbf{J}(\xi)(\mathbf{x}_n - \boldsymbol{\alpha}) \text{ where } \xi = \mathbf{x}_n + h(\boldsymbol{\alpha} - \mathbf{x}_n),\ h \in [0,1] \\ &= \lambda_n(\mathbf{x}_n - \alpha) \text{ when } \lambda_n \text{ is an eigen value of } \mathbf{J}(\xi) \end{aligned}$$

So, this must be lesser than or equal to m times norm of x minus y. Thus we can say that norm of phi of x is actually called is not really norm of phi of x. Phi of x is a contraction mapping. Why is it a contraction mapping? Well, let us take a look. What it does is that if we have x and y, if I have two members belonging to that neighborhood, two points to vectors, let us take it as points which belong to that neighborhood r centered at alpha. See if I operate with phi on x and if I operate with phi on y, then the norm of phi x minus phi y, we can think of the norm as the distance norm is always like some sort of distance, right. So, the norm of phi x minus phi y is lesser than or equal to m. Remember m is less than 1.

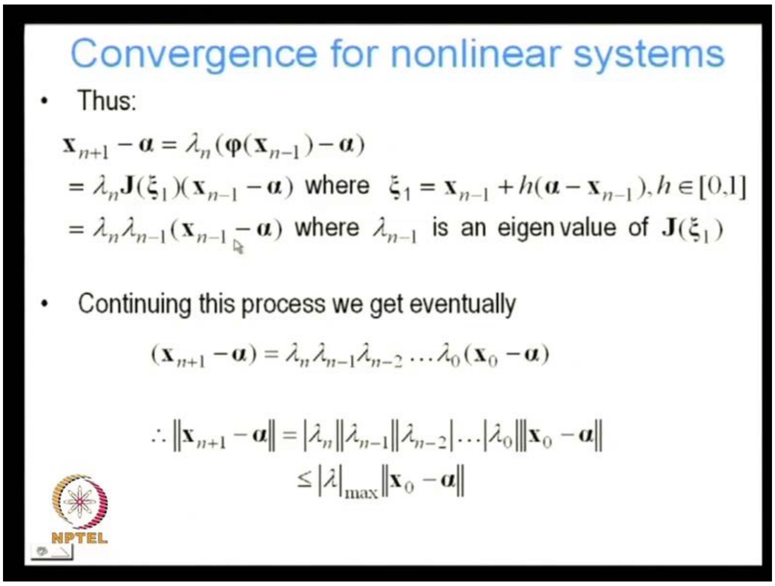
So, is less than the distance between x and y in the same node, right. So, originally I had points x and y. If I look at the distance between x and y norm is like if I think of norm is like distance. I compute the distance between x and y, then I operate on x with phi, I get phi of x I, operate on y with phi, I get phi of y and when after the transformation, I compute again the distance between the results of the transformation and I will see that after my transformation, the distance between those things is actually less than my original distance, right.

So, it is an contraction mapping. The distance is becoming smaller, right. Distance of the mapped variable is smaller than my original variable. So, that is the contraction mapping. Since, the norm of the vector between the mapped values phi of x minus phi of y is strictly smaller than the norm of the vector between the original values x and y, right. That is why it is the contraction, a necessary condition for x k plus 1 is equal to phi of x k. That is the iteration algorithm to converge is the spectral radius of J x, the spectral radius.

What is the spectral? It has the largest Eigen value of J x for any x belonging to r neighborhood alpha centered at alpha with radius row is less than or equal to 1. Well, we can see this like we can prove this as follows. Let us see. Let us write x n plus 1 minus alpha n plus 1 is the value of the iterate minus the root. So, x n plus 1 is equal to phi of x n alpha is equal to phi of alpha because alpha is the root, right, so phi of x n minus phi of alpha.

Again, let us use our mean value theorem and we can write J of psi is equal to x n minus alpha, where psi is equal to now x n plus h times alpha minus x n h belongs to 0, and 1. This I can write as lambda n times x n minus alpha when lambda n is an Eigen value of J psi. So, this is the just a statement, standard statement of the Eigen value problem. This is the matrix that is my sort of Eigen vector that is my Eigen value, right. That is an Eigen value. So, I can write this like this, right.

(Refer Slide Time: 16:11)



Convergence for nonlinear systems

- Thus:

$$\mathbf{x}_{n+1}-\boldsymbol{\alpha} = \lambda_n(\boldsymbol{\varphi}(\mathbf{x}_{n-1})-\boldsymbol{\alpha})$$

$= \lambda_n \mathbf{J}(\xi_1)(\mathbf{x}_{n-1}-\boldsymbol{\alpha})$ where $\xi_1 = \mathbf{x}_{n-1}+h(\boldsymbol{\alpha}-\mathbf{x}_{n-1}), h\in[0,1]$

$= \lambda_n\lambda_{n-1}(\mathbf{x}_{n-1}-\boldsymbol{\alpha})$ where λ_{n-1} is an eigen value of $\mathbf{J}(\xi_1)$

- Continuing this process we get eventually

$$(\mathbf{x}_{n+1}-\boldsymbol{\alpha}) = \lambda_n\lambda_{n-1}\lambda_{n-2}\dots\lambda_0(\mathbf{x}_0-\boldsymbol{\alpha})$$

$$\therefore \|\mathbf{x}_{n+1}-\boldsymbol{\alpha}\| = |\lambda_n||\lambda_{n-1}||\lambda_{n-2}|\dots|\lambda_0|\|\mathbf{x}_0-\boldsymbol{\alpha}\|$$

$$\le |\lambda|_{\max}\|\mathbf{x}_0-\boldsymbol{\alpha}\|$$

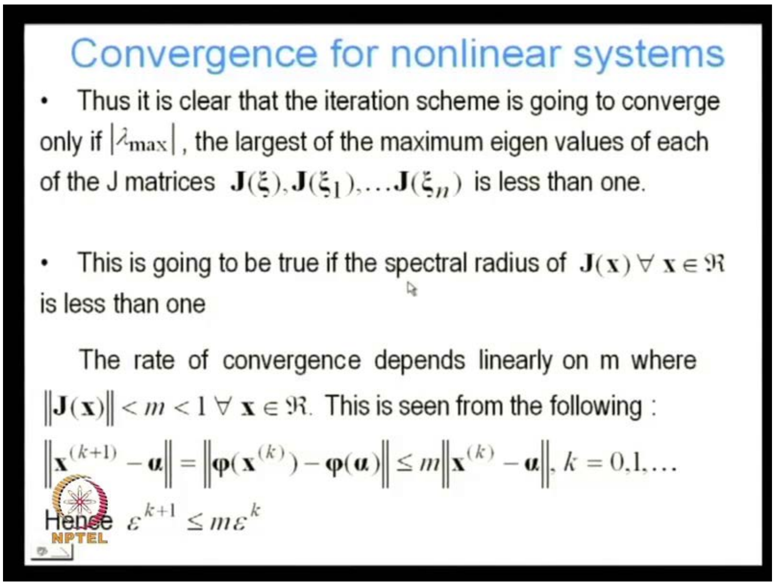
So, what do we have? We have x n plus 1 minus alpha is equal to lambda n times x n minus alpha, right. So, we had x n minus alpha. So, I can write x n as phi of x n minus 1 minus alpha and we have done this so many times that this should be familiar to you now. So, now, I can write lambda n phi of x n minus alpha. I will again use the same mean value theorem.

So, I can write this as J of psi one times x n minus alpha, where now psi 1 belongs to the interval between x n minus 1 and alpha. So, again x n minus 1 plus h times alpha minus x n minus 1 h belongs to 0 1. So, now, this again can be if I write it as I can write it in terms of Eigen values and Eigen vectors. So, J of psi 1 times x n minus 1 minus alpha, that becomes lambda n minus 1 x n minus 1 alpha, where lambda n minus 1 is a Eigen value of J psi 1 x n minus alpha. So, now, I have lambda n minus 1 x n minus 1 alpha where lambda n minus 1 is an Eigen value of this.

So, continuing this process, again I am going to write x minus 1 as psi of x n minus 2 minus alpha. Again I am going to use the mean value theorem, and I am going to write it in terms of Eigen values and Eigen vectors. Eventually, we are going to get lambda n, lambda n minus 1, lambda n minus 2 through lambda 0 times x 0 minus alpha, right. This is my initial iterative, my initial guess to the value of the root. So, this is what I get and then I take norm on the both sides.

I take the norm on both sides and I get lambda, the absolute value of these Eigen values, right and I have norm of x 0 minus alpha, right and this is always going to be less than the maximum of those Eigen values, right. So, I look at each. This is like I am looking at all those iterations, right. So, I look at the maximum Eigen value of each iteration, right and that maximum Eigen value over all the iterations if I replace that here, I am sorry this should be a power n, right. So, this should be n, right and this is norm of x 0 minus alpha, right. So, this to the power n times, this is going to be greater than or equal to that, right.

(Refer Slide Time: 18:51)



Convergence for nonlinear systems

- Thus it is clear that the iteration scheme is going to converge only if $|\lambda_{\max}|$, the largest of the maximum eigen values of each of the J matrices $\mathbf{J}(\xi), \mathbf{J}(\xi_1), \ldots \mathbf{J}(\xi_n)$ is less than one.

- This is going to be true if the spectral radius of $\mathbf{J}(\mathbf{x}) \, \forall \, \mathbf{x} \in \Re$ is less than one

The rate of convergence depends linearly on m where $\|\mathbf{J}(\mathbf{x})\| < m < 1 \, \forall \, \mathbf{x} \in \Re$. This is seen from the following :

$$\left\|\mathbf{x}^{(k+1)} - \boldsymbol{\alpha}\right\| = \left\|\boldsymbol{\varphi}(\mathbf{x}^{(k)}) - \boldsymbol{\varphi}(\boldsymbol{\alpha})\right\| \leq m\left\|\mathbf{x}^{(k)} - \boldsymbol{\alpha}\right\|, \; k = 0,1,\ldots$$

Hence $\varepsilon^{k+1} \leq m\varepsilon^{k}$

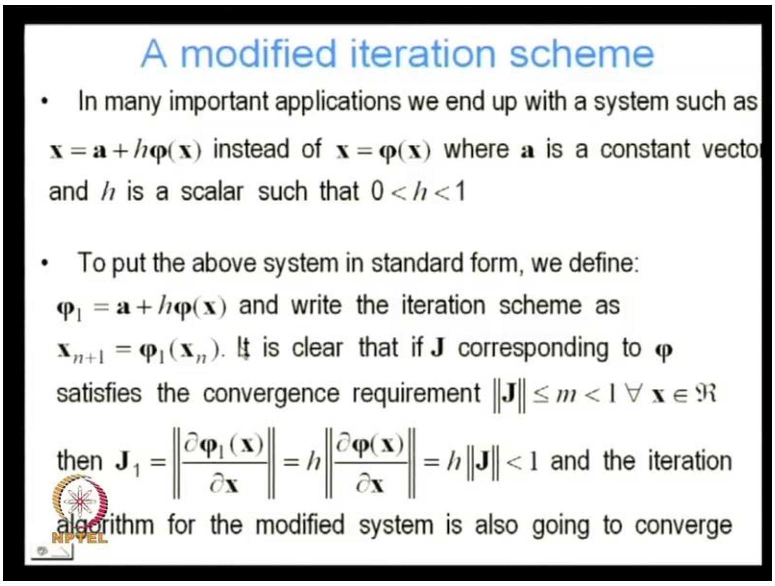
So, thus it is clear. The iteration scheme is going to converge only if mod of lambda max, the largest of the maximum Eigen values of each of the J matrices J psi, J psi 1, J psi n is less than 1, right. So, the maximum of maximum Eigen value of this matrix and then the maximum of those if is less than 1, then we are going to get convergence, right. So, what does this mean? That means, that the spectral radius of J x for x belonging to r because this is going to be true for all these psi, right and only what is common between psi, psi 1, psi 2, psi 3 and psi n is they all belong to r, right. So, they can be anywhere in r, right. So, that means that for this to converge for any point x belonging to r, J of x must have maximum Eigen value less than 1, right. So, that is going to be true if the spectral radius of J of x is less than 1. So, that was our convergence criteria.

Let us look at rate of convergence, right. Remember we looked at the rate of convergence for the one-dimensional non-linear equation in turn Raphson and so on and so forth. So, what can we say about the rate of convergence of this is that the rate of convergence depends linearly on m, where norm of g of x is less than m is less than 1 for all x belonging to r.

How can we see this? Well, let us take a look. So, norm of x k plus 1 minus alpha, I can write it as norm of phi of x k minus phi of alpha which is equal to lesser than or equal to m times norm of x k plus 1 minus alpha. I can write it as norm of phi of x k minus phi of alpha which is equal to lesser than or equal to m times norm of x k minus alpha. We showed that earlier around, right. Let us take a look at that again from this, right. We showed that earlier, right.

So, this is going to be lesser than or equal to norm of x k minus alpha. What is this? It is epsilon k plus 1. It is the error at the k plus 1 with the iteration because it is the value of the iterate minus the value of the iterate at k plus 1 minus the root at minus the root. Similarly, this is the error at k because this is the value of the iterate minus the root. So, we have epsilon k plus 1 is lesser than or equal to m times epsilon k. So, whatever is the value of m that is going to cover my convergence. So, that was that.

(Refer Slide Time: 21:48)



A modified iteration scheme

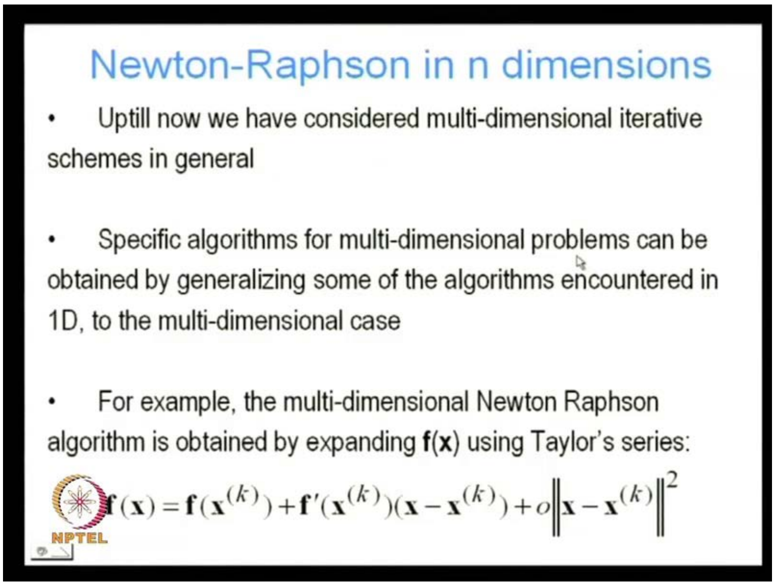
- In many important applications we end up with a system such as $\mathbf{x} = \mathbf{a} + h\boldsymbol{\varphi}(\mathbf{x})$ instead of $\mathbf{x} = \boldsymbol{\varphi}(\mathbf{x})$ where $\mathbf{a}$ is a constant vecto and h is a scalar such that $0 < h < 1$
- To put the above system in standard form, we define: $\boldsymbol{\varphi}_1 = \mathbf{a} + h\boldsymbol{\varphi}(\mathbf{x})$ and write the iteration scheme as $\mathbf{x}_{n+1} = \boldsymbol{\varphi}_1(\mathbf{x}_n)$. It is clear that if $\mathbf{J}$ corresponding to $\boldsymbol{\varphi}$ satisfies the convergence requirement $\|\mathbf{J}\| \le m < 1 \ \forall \ \mathbf{x} \in \Re$ then $\mathbf{J}_1 = \left\| \frac{\partial \boldsymbol{\varphi}_1(\mathbf{x})}{\partial \mathbf{x}} \right\| = h \left\| \frac{\partial \boldsymbol{\varphi}(\mathbf{x})}{\partial \mathbf{x}} \right\| = h\|\mathbf{J}\| < 1$ and the iteration algorithm for the modified system is also going to converge

So, now, let us look at that was the general iteration scheme for phi of x. X is equal to x, right. So, now, often instead of having phi x is equal to phi of x we have because we are modeling. After all in civil engineering or any engineering, we do not solve equations for the sake of solving questions. We are trying to model physical phenomenon, right. It may so happen that my physical phenomenon gives me an equation like this, right. X is equal to a plus h of phi of x instead of x is equal to phi of x, where x is a constant vector and h is a scalar, such that h is greater than or equal to 0 greater than 0 and less than 1. We can very easily put this system into a standard form.

How do we do that? We have to put it in the standard form. We define phi 1 which is equal to a plus h of phi of x and write the iteration scheme as x n plus 1 is equal to phi 1 of x n. It is clear that if J corresponding to this, corresponding to phi satisfies the convergence requirement norm of J lesser than or equal to J less than 1, for all x belonging to r J 1, which actually corresponds to that is also going to satisfy that requirement. How well J 1 is equal to norm of J 1? It is not J 1. I am sorry, norm of J 1 is equal to norm of del phi 1 del x, which is equal to if we look at this, right.

What is norm of del phi 1 del x is nothing but a h times norm of del phi del x, right and this is going to be a h times norm of J and since, this is less than 1 and the iteration algorithm for this is also going to converge because norm of J 1 is also going to be less than 1, right. So, if norm of J is less than 1, norm of J 1 is also going to be less than 1 and the modified algorithm is also going to converge. So, in case of simple transformation, simple I mean it is slightly different. Iterative scheme that also we are guaranteed is going to converge provided that this condition is satisfied, right.

(Refer Slide Time: 24:26)



Newton-Raphson in n dimensions

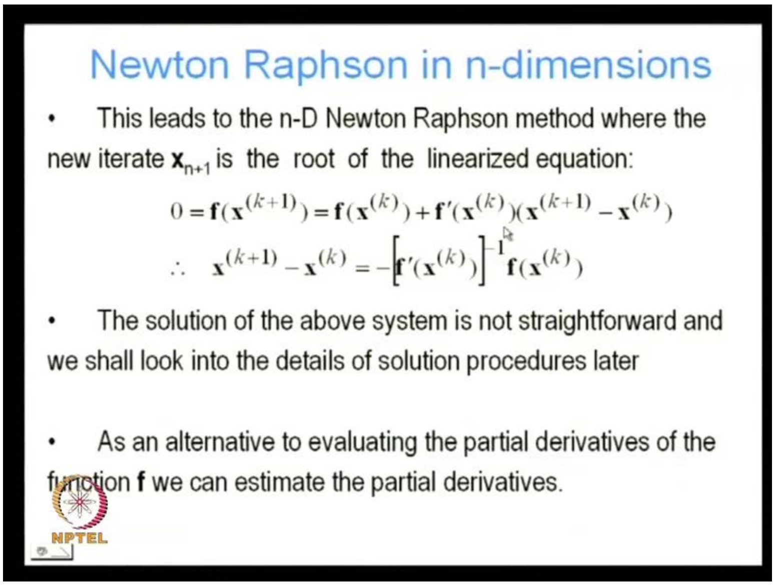
- Uptill now we have considered multi-dimensional iterative schemes in general
- Specific algorithms for multi-dimensional problems can be obtained by generalizing some of the algorithms encountered in 1D, to the multi-dimensional case
- For example, the multi-dimensional Newton Raphson algorithm is obtained by expanding **f**(**x**) using Taylor's series:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}^{(k)}) + \mathbf{f}'(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) + o\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|^2$$

So, up till now we talked about general algorithms, and we talked about general convergence characteristics, convergence requirements, grades of convergence. Let us become slightly more specific and look at certain specific algorithms, right. The first algorithm we look at will be Newton Raphson, right. That is the most commonly used algorithm, most widely used algorithm, most versatile.

So, the Newton Raphson method in order to do that, in order to look at in greater detail what we need to do is to see how we get it in multi-dimensions right and to see how we get it in multi-dimensions, let us try to expand this function f of x which remember is now vector, right. It is f 1, it is been f 1, f 2, f 3, f n. So, function of x 1, x 2, x 3, x n and let us expand it in multi-dimensional Taylor series, right and let us do that f of x is equal to f of x k. So, I am expanding it about iterate at step k. So, f of x k plus f prime evaluated at x k plus x minus x k plus the correction term, the remainder term, the higher order term which is going to be quadratic in high, right.

(Refer Slide Time: 26:03)



Newton Raphson in n-dimensions

- This leads to the n-D Newton Raphson method where the new iterate $\mathbf{x}_{n+1}$ is the root of the linearized equation:

$$0 = \mathbf{f}(\mathbf{x}^{(k+1)}) = \mathbf{f}(\mathbf{x}^{(k)}) + \mathbf{f}'(\mathbf{x}^{(k)})(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$$

$$\therefore \quad \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = -\left[\mathbf{f}'(\mathbf{x}^{(k)})\right]^{-1}\mathbf{f}(\mathbf{x}^{(k)})$$

- The solution of the above system is not straightforward and we shall look into the details of solution procedures later

- As an alternative to evaluating the partial derivatives of the function **f** we can estimate the partial derivatives.

So, this is my Taylor series expansion and this leads to the n-dimensional Newton Raphson method, where the new iterate x n plus 1 is the root of the linearized equation. Remember how did we get our Newton Raphson method in one-dimensions? We linearized it, we used the Taylor series expansion, but then we cut it off before the quadratic term, right.

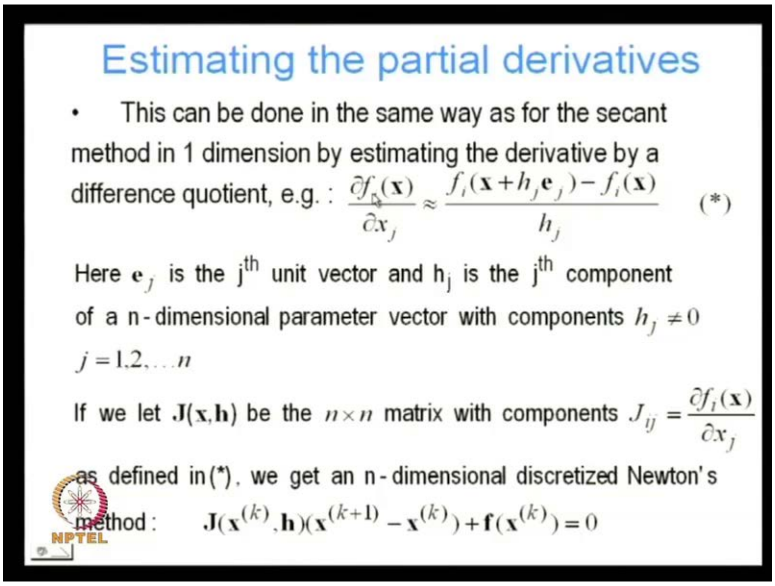
We kept only up to the linear term. So, we linearized our expression, right and that led us to the Newton Raphson algorithm. So, let us do the same thing in multi-dimensions. So, what do we do? We say that f of x k plus 1 is equal to f of x k plus f prime. This we already saw and we say that at my new iterate x k plus 1, remember what we did in one-dimensional Newton Raphson. We took the tangent from the point.

What is my new iterate? My new iterate is where my tangent intersects the x axis, right. What does that mean? That means that the function value I am looking, assuming that the function value is 0 because it is intersecting. It has got ordinate 0, it is intersecting the x axis, right. So, what I am saying is that my x k plus 1 f of x k plus 1 is equal to 0 because that is where it is intersecting my x axis, right. So, it is just the parallel to that, right. So, my new iterate I find it like this, right. So, it is f of x k plus 1. So, at x k plus 1, at the new value of my iterate, new value of my iterate. Remember in 1D was where it intersected the x axis, right. When would it intersect? When would a function intersect x axis is when the function value is 0, right. So, similarly I am assuming that the function value is 0, right. So, in that case this becomes 0 and I can write x k plus 1 minus x k is equal to this.

Now, remember this is a matrix, right. This is Jacobian matrix. So, this inverse times f of x k, right. So, that gives me my new iterate x k plus 1, right. So, same thing generalize same idea as extended to multi-dimensions, right. So, the assumption is that x k. So, the assumption is that if I draw, if I take a point on my solution and I draw my tangent and assume that is the root of that equation, that is f of x k plus 1 is equal to 0, right. In that case, that gives me my new iterative according to Newton’s method, right.

So, the solution of this system is not straight forward at all, right and we shall look into the details of the solution procedure later, but one thing to note is that evaluating these partial derivatives. So, what does Newton algorithm in multi-dimensions require? It means that at every iterate, at every iteration I have to evaluate that n by n matrix of partial derivatives, right. I not only do have to evaluate, I have to invert it also, so that the evaluation of the derivative is expensive, right. As an alternative to evaluating the partial derivatives of the function f, actually evaluating the partial derivatives, we can estimate the partial derivatives, right. How do we do that? Well, let us take a look.

(Refer Slide Time: 29:49)



Estimating the partial derivatives

- This can be done in the same way as for the secant method in 1 dimension by estimating the derivative by a difference quotient, e.g. : $\frac{\partial f_i(\mathbf{x})}{\partial x_j} \approx \frac{f_i(\mathbf{x}+h_j\mathbf{e}_j)-f_i(\mathbf{x})}{h_j}$ (*)

Here $\mathbf{e}_j$ is the j^{th} unit vector and h_j is the j^{th} component of a n-dimensional parameter vector with components $h_j \neq 0$ $j = 1,2,\ldots n$

If we let $\mathbf{J}(\mathbf{x},\mathbf{h})$ be the $n \times n$ matrix with components $J_{ij} = \frac{\partial f_i(\mathbf{x})}{\partial x_j}$ as defined in (*), we get an n-dimensional discretized Newton's method : $\mathbf{J}(\mathbf{x}^{(k)},\mathbf{h})(\mathbf{x}^{(k+1)}-\mathbf{x}^{(k)})+\mathbf{f}(\mathbf{x}^{(k)})=0$

This can be done in the same way as this. Remember we looked at the secret method in 1D and when we started looking at the secret method, we started with the Newton Raphson method and said that the second method comes about if we instead of evaluating the actual derivative, we construct an approximation to the derivative, right. We construct an approximation to the derivative. How do we construct the approximation to the derivative? It is by evaluating the function value at two points, and dividing it by the distance between those two points, right. That is an approximation to my derivative.

So, we were going to do the same thing in case of multi-dimensions, but in this case we are going to approximate each of those little partial derivatives in major Jacobian matrix. How am I going to do that? Well, I am going to say that my del phi del f y del x J, my partial derivative of function f i with respect to the independent variable x J, I can say that is approximately equal to f i x plus h J times e j, right. What is e j? E j is nothing but the unit vector in the J direction.

So, if you think of in 3D, if we are looking at f 3, it is x plus h 3 times e 3. So, e 3 has got 0, first two terms is 0 and the last one is 1, right. So, it is the h. It is an increment in the jth component of my vector, right. If I think of an n dimensional vector since this is an n dimensional space, if I think of an n dimensional vector, I am looking at an increment in the nth component of that vector in the nth component of that vector.

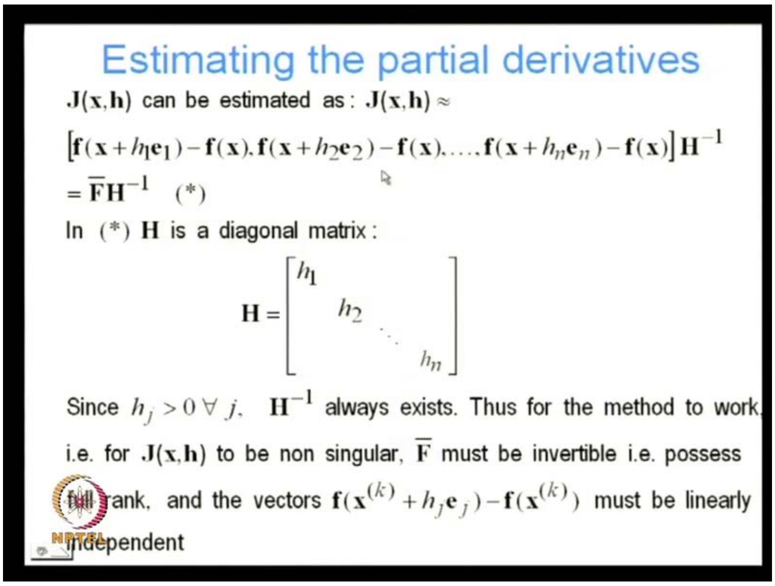
What is the magnitude of that increment? The magnitude of that increment is h n if I look at the jth component of the vector. So, every other component is 0 except the jth component. So, it is the unit vector in the jth unit vector in my n dimensional space, right and that jth unit vector I am scaling it with a value h j, right. So, one can think of it like I am starting from x and I am moving in the direction of the jth unit vector.

How much I am moving it? I am moving by a step of size h j, right. I am taking a step of size h j, right, so f i x plus h j e j minus f i of x. So, that is like this. This is just I am trying to talk, I am trying to explain what basically in mathematical term this is known as directional derivative, right. It is basically a directional derivative, right. Since, directional derivative of f in the jth direction, it is directional derivative of f i x plus h j e j minus f i of x divided by h of j, right.

So, that is how I am approximating this by its directional derivative e j is the jth unit vector, and h j is the jth component of n dimensional parameter vector with components h j not equal to 0 j is equal to 1 to n. So, I am thinking of all these h as one vector and each component of that vector is a parameter, and is less than 1. So, if we let j x h be the n by n matrix with components J i J which are defined like this, so these are my components in my J matrix.

Now, you can see that it has become a function not only of x, but also of those h, right. So, J i J if we define like that, we get an n dimensional discretised Newton method, right. Now, I am no longer evaluating the exact derivatives, but I am evaluating it like this, but the structure remains the same, J x k h times x k plus 1 minus x k plus f of x k equal to 0. Look at that here, right. So, that was what we had earlier around. Only difference is now instead of evaluating this derivative, exactly I am evaluating it like this, right.

(Refer Slide Time: 34:35)



Estimating the partial derivatives

$\mathbf{J}(\mathbf{x},\mathbf{h})$ can be estimated as: $\mathbf{J}(\mathbf{x},\mathbf{h}) \approx$

$$\left[\mathbf{f}(\mathbf{x}+h_1\mathbf{e}_1)-\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}+h_2\mathbf{e}_2)-\mathbf{f}(\mathbf{x}), \ldots, \mathbf{f}(\mathbf{x}+h_n\mathbf{e}_n)-\mathbf{f}(\mathbf{x})\right]\mathbf{H}^{-1}$$

$$= \overline{\mathbf{F}}\mathbf{H}^{-1} \quad (*)$$

In $(*)$ $\mathbf{H}$ is a diagonal matrix:

$$\mathbf{H} = \begin{bmatrix} h_1 & & & \\ & h_2 & & \\ & & \ddots & \\ & & & h_n \end{bmatrix}$$

Since $h_j > 0 \; \forall \; j$, $\mathbf{H}^{-1}$ always exists. Thus for the method to work, i.e. for $\mathbf{J}(\mathbf{x},\mathbf{h})$ to be non singular, $\overline{\mathbf{F}}$ must be invertible i.e. possess full rank, and the vectors $\mathbf{f}(\mathbf{x}^{(k)}+h_j\mathbf{e}_j)-\mathbf{f}(\mathbf{x}^{(k)})$ must be linearly independent

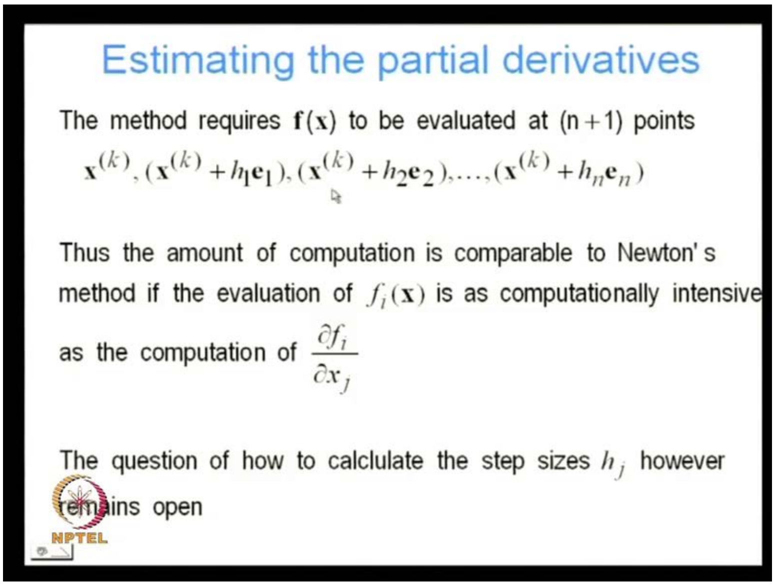
So, if we do that, so J of x h can be approximated as, so we can write this as basically let me go back again. So, I am going to write this as a matrix of these vectors. So, basically it is this all these vectors, right. So, all these vectors f of x plus h 1 e 1 minus f of x f of x plus h 2 e 2 minus f of x and so on and so forth, right. We can see why that is true, right. So, f 1 x 1, right. So, that is going to be f 1 x plus h 1 e 1 minus f 1 of x by h 1, right.

So, let us take a look that is f 1. So, that is going to f 1 x plus h 1 e 1 minus f of x and then next row will be f 2 x plus h 1 e 1 minus f of x and the first row first column will be f 1 x plus h 2 e 2 minus f of x. All those derivatives for all those partial derivatives I am going to get a term like this, right and I have to divide by these h s. So, I am taking those h s outside, right and writing it as a diagonal matrix.

So, this equation is identical. It is just I am writing this equation, right. I am writing it in matrix form, right. If I write it in matrix form, I get f bar times h inverse, right. Let h is a diagonal matrix which contains all the h s, all these increment sizes on its diagonal terms since all these h j are greater than 0, then h inverse always exist. It is a diagonal matrix with non 0 diagonal terms which inverse must exist, right. Thus for the method to work, but remember what do we have to do. What do we have to do to find my increment? I have to invert this matrix, right. So, to invert this matrix, not only must this inverse exist, this be invertible. This must also be invertible. When that is going to be invertible, then it has got full rank.

What does full rank mean? That means, all these columns, right. All these columns f 1 x plus h 1 e 1 minus f 1 x f 2 x plus h 1 e 1 minus f 1 x and so on and so forth and this column, all these columns are linearly independent. In that case, it is going to have full rank, right. It is going to have full rank and then this Jacobian is going to be invertible and it is going to be that my Newton algorithm is going to work.

(Refer Slide Time: 37:34)



Estimating the partial derivatives

The method requires $\mathbf{f}(\mathbf{x})$ to be evaluated at (n+1) points

$$\mathbf{x}^{(k)}, (\mathbf{x}^{(k)} + h_1\mathbf{e}_1), (\mathbf{x}^{(k)} + h_2\mathbf{e}_2), \ldots, (\mathbf{x}^{(k)} + h_n\mathbf{e}_n)$$

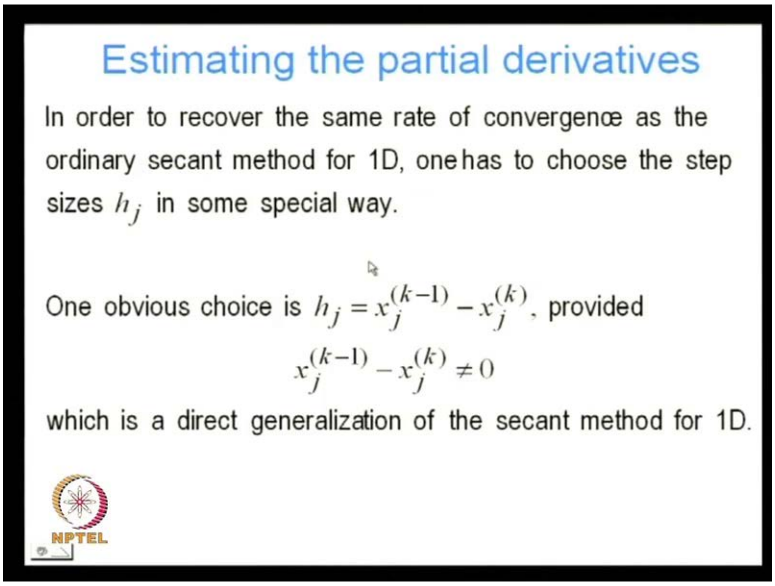
Thus the amount of computation is comparable to Newton' s method if the evaluation of $f_i(\mathbf{x})$ is as computationally intensive as the computation of $\frac{\partial f_i}{\partial x_j}$

The question of how to calclulate the step sizes h_j however remains open

For this method to work, we have to evaluate this function at n plus 1 point, right. So, you can see all these functions have to be evaluated at each point, right. So, we have do lot more function evaluations, right. So, it has to be evaluated at x k x plus h 1 e 1 x k plus h 2 e 2 and so at any iteration, I have to do all these evaluations, right. Any iterate k I have to do all these function evaluations, function has to be evaluated at all these values. Thus, the amount of these function evaluations are to evaluate the function is almost as expensive as evaluating these derivatives. Then it is not much use, right. Then I can just do my usual full Newton derivatives, right.

So, if the value, if the amount of computation is comparable to Newton's method, if the evaluation of f i x is as computationally intensive as the computation of this partial derivatives, but the question is how do we calculate it. So, we have done everything and it works fine. We have worked. We have seen it works fine. So, as long as these columns are linearly dependent and all those h s are greater than 0, but then what should I take my h s to be. I know that they have to be greater than 0 for it to be invertible, but then how much should that size be. That is depending on what I take my h s to be. I am going to get fast or slow convergence, right. You can assume, you can think that if I take very large steps, it is a non-linear function. If I take very large h j, then I am not going to get good convergence because the function is bearing non-linearly and if I take very large h j to estimate my derivatives, my derivatives are not going to be accurate, right.

(Refer Slide Time: 39:48)



Estimating the partial derivatives

In order to recover the same rate of convergence as the ordinary secant method for 1D, one has to choose the step sizes h_j in some special way.

One obvious choice is $h_j = x_j^{(k-1)} - x_j^{(k)}$, provided

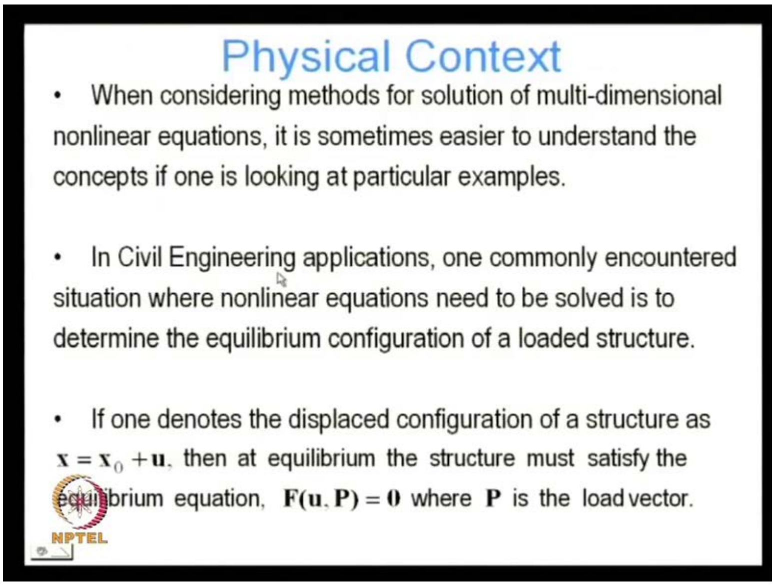
$$x_j^{(k-1)} - x_j^{(k)} \neq 0$$

which is a direct generalization of the secant method for 1D.

So, then what should I use my h j s to be? Well, there can be lots of ways to chose h j, but one way to chose it is use my secant algorithm secant way. I chose my h j for my secant algorithm, right. What do I chose my h j to be? It is just the previous iterate minus the value of the x J component of x, the previous iterate value minus the current iterate value, right. So, that is one possible choice of my h j s depending on how I chose my h j s. I am going to get my rate of convergence is going to be different, right and if I use my, if my chose my h j s like I do for my secant method, I am going to get the same rate of convergence as my ordinary secant method.

Remember that now we can no longer expect to get quadratic convergence. Why? Because we are not choosing, we are not using the exact derivatives because we are approximating these derivatives using this sort of directional derivatives, right. So, now, we can no longer get quadratic convergence. Order of convergence is going to be less than 2, but if we chose the h j like it is chosen for secant method, then I am assured that I am going to get at least as higher rate of convergence as the secant method, and we saw that the rate of convergence for the secant method for something like 1.6, right. So, instead of getting quadratic convergence equal to 2, I am going to get order of convergence like 1.6 something.

(Refer Slide Time: 41:30)



Physical Context

- When considering methods for solution of multi-dimensional nonlinear equations, it is sometimes easier to understand the concepts if one is looking at particular examples.

- In Civil Engineering applications, one commonly encountered situation where nonlinear equations need to be solved is to determine the equilibrium configuration of a loaded structure.

- If one denotes the displaced configuration of a structure as $\mathbf{x} = \mathbf{x}_0 + \mathbf{u}$, then at equilibrium the structure must satisfy the equilibrium equation, $\mathbf{F}(\mathbf{u}, \mathbf{P}) = \mathbf{0}$ where $\mathbf{P}$ is the load vector.

So, because we said that we are going to talk in greater detail about ways to solve this equation on ways to solve and where have we skipped that anyway. We said that we are going to talk of ways to solve that in that equation, right. This equation and not this equation, right. So, let us do that, but when we look at ways of solving that equation, it often makes sense to look at particular context because it has become quite complicated. So, it makes a lot of sense, it is much easier to understand if we look at physical problem, right. We say that we are solving. Look at how we solve this equation in the context of a physical problem, right. So, we are going to look at multi-dimensional Newton algorithm convergence and things like that.

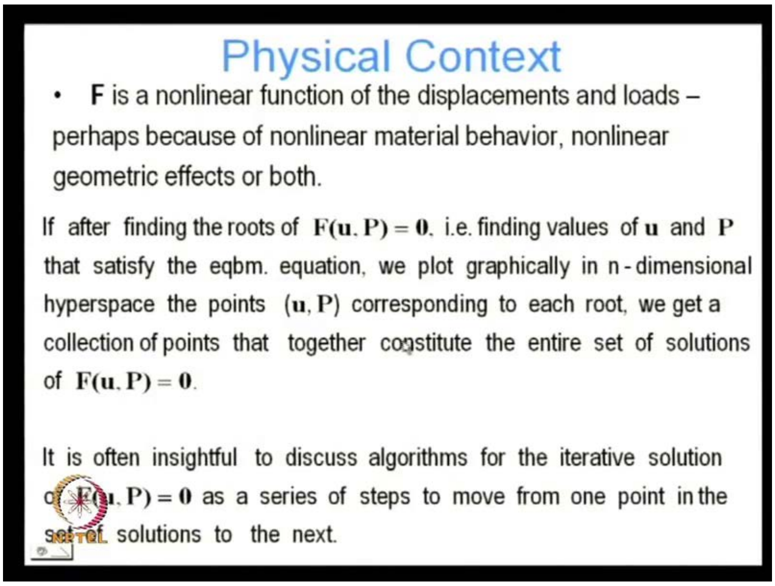
In a physical context and in civil engineering applications, one commonly encountered context for solving non-linear equations is when we want to determine the equilibrium configuration of a loaded structure, provided that loaded structure is behaving in a non-linear fashion, right. My equation in this case which becomes the equation of motion and put it more generally the equation of conservation of momentum, right.

That equation is non-linear, right. Why is it non-linear? Well, it can be non-linear because of several reasons. It can be because my structure has made of a non-linear material, right or because I have finite deformations, right. So, my displacement strain relations, I do not want to get too much into the mechanics aspect of the things, but they can be lot of reasons why my equation is non-linear.

So, now I want to solve that equilibrium configuration of a loaded structure. So, to do that we denote the displaced configuration of a structure as x is equal to x 0 plus u. Suppose my structure and initial configuration, it had configuration x 0 and then I provided displacement u and this gives me my displaced configuration of the structure. So, for that structure to be in equilibrium, it must satisfy this equation f of u p is equal to 0, where f is if u p is my equilibrium equation and it has variable u which is basically representing my displacement, but we can think of it as where f of x, my displace configuration, but since x o is a constant as I am assuming that my initial configuration is known. So, I can write it as a function of u and p, where p is my load vector.

So, for to solve a non-linear problem, basically one can think of that as finding the roots of that equation, finding the root of f of u p is equal to 0. I want to find displaced configuration and load which satisfies my equilibrium equation, right. My equilibrium equation is f of u p. So, I want to find root of that n dimensional non-linear equation. What are those roots? Roots are my displacements in n dimensional displacements, n vectors, and my load which is also a n dimensional vector, right. So, find u and p such that f of u p is equal to 0.

(Refer Slide Time: 45:36)



Physical Context

- **F** is a nonlinear function of the displacements and loads – perhaps because of nonlinear material behavior, nonlinear geometric effects or both.

If after finding the roots of $\mathbf{F}(\mathbf{u},\mathbf{P}) = \mathbf{0}$, i.e. finding values of $\mathbf{u}$ and $\mathbf{P}$ that satisfy the eqbm. equation, we plot graphically in n - dimensional hyperspace the points $(\mathbf{u},\mathbf{P})$ corresponding to each root, we get a collection of points that together constitute the entire set of solutions of $\mathbf{F}(\mathbf{u},\mathbf{P}) = \mathbf{0}$.

It is often insightful to discuss algorithms for the iterative solution of $\mathbf{F}(\mathbf{u},\mathbf{P}) = \mathbf{0}$ as a series of steps to move from one point in the set of solutions to the next.

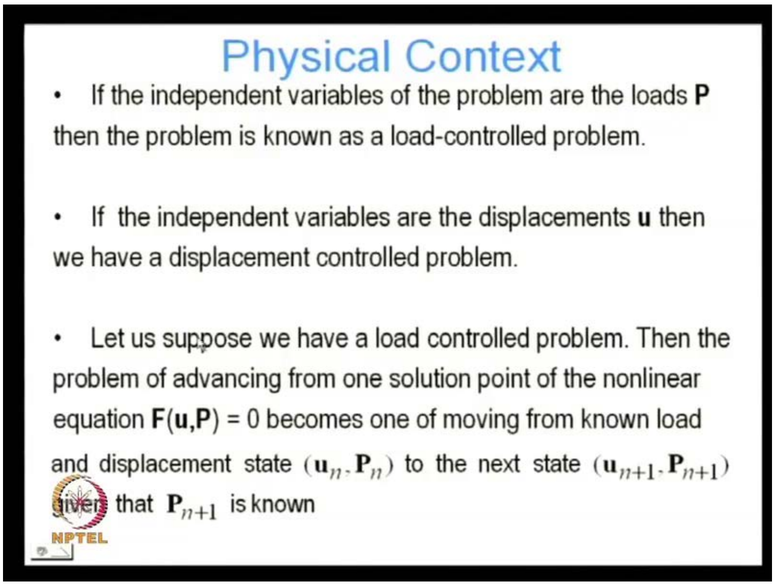
F is a non-linear function of the displacements and loads which may be non-linear because may be non-linear material behavior, non-linear geometric we do not want to concern to assess too much with that, but because of some non-linearity, let us suppose, right. So, if after the findings of F, finding the roots of F of u p is equal to 0 that is after finding the values of u and p that satisfy the equilibrium equation, we plot graphically in n dimensional hyper space, right. So, we have an n dimensional hyper space and for each dimension correspond to one particular value of u, one component of u, right and may be one component of another dimension corresponds to another component of u. Another dimension might correspond to one component of p, right.

So, when n dimensional hyper space and in that n dimensional hyper space I have points, I plot the points. What are those points? Each point corresponds to a value of u and a value of p which is a root of my equilibrium equation, right. So, I plot in my n dimensional hyper space, all those points 3D we can think points at a surface, right. So, we get a collection of points that together constitute the entire set of solutions of f of u p is equal to 0, right. So, it is often insightful to discuss algorithms for the iterative solution of f u p as a series of steps to move from one point in the set of solutions to the next.

So, now, I have all these points in the n dimensional hyper space and the basic problem of solving this non-linear equation is that given a point, given suppose a starting point, given one of those points, one of those infinite number of points, given one of those points, how do I move to the next point and what is driving that. Well, one can drive it either by changing the displacements or by changing the loads. If its displacements driven problem I know my initial displacement and my initial load at an initial equilibrium configuration equation, then I change the displacement, right. I change u n to u n plus delta, right.

So, change my u at delta by delta and then I want to find what is my p which is going to satisfy F of u p is equal to 0 that is going to be true if its displacement driven problem, if its load driven problem, it is just the other way wrong. So, again I have an equilibrium configuration where all my displacements and all my u s and all my p s are known, right and then I change my load. I change my p by delta p, right and then I want to find what value of u, how must I change u to satisfy the equilibrium equation, right.

(Refer Slide Time: 48:53)

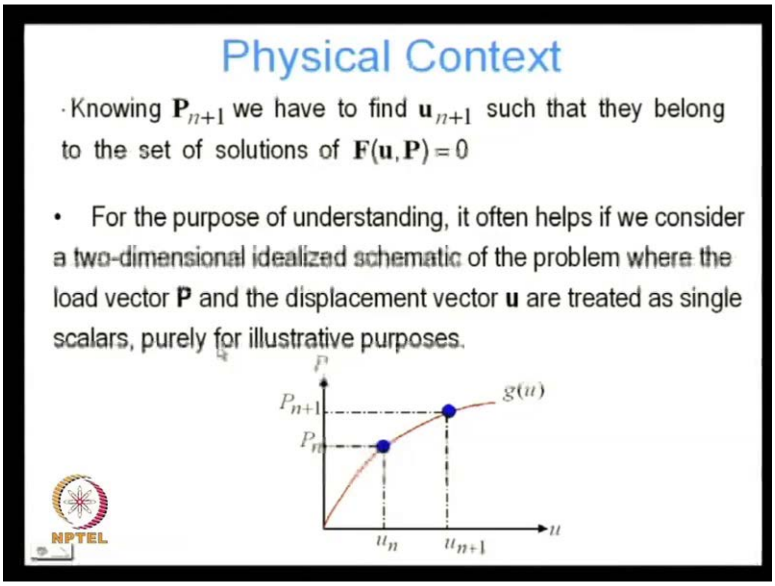


Physical Context

- If the independent variables of the problem are the loads **P** then the problem is known as a load-controlled problem.
- If the independent variables are the displacements **u** then we have a displacement controlled problem.
- Let us suppose we have a load controlled problem. Then the problem of advancing from one solution point of the nonlinear equation **F**(**u**,**P**) = 0 becomes one of moving from known load and displacement state $(\mathbf{u}_n, \mathbf{P}_n)$ to the next state $(\mathbf{u}_{n+1}, \mathbf{P}_{n+1})$ given that $\mathbf{P}_{n+1}$ is known

If the independent variables of the problem are the loads p, then the problem is a load controlled problem. If the independent variables are the displacements u, then we have a displacement controlled problem. Let us suppose we have a load controlled problem, then the problem of advancing from one solution point of the non-linear equation F u p equal to 0 becomes one of moving from known load, and displacement state u n p n to the next state u n plus 1 p n plus 1 p n plus 1 is known since it is a load controlled problem. I know my p n plus 1. I want to find u n plus 1, such that my equilibrium equation is satisfied.

(Refer Slide Time: 49:44)



Physical Context

- Knowing $\mathbf{P}_{n+1}$ we have to find $\mathbf{u}_{n+1}$ such that they belong to the set of solutions of $\mathbf{F}(\mathbf{u},\mathbf{P})=0$

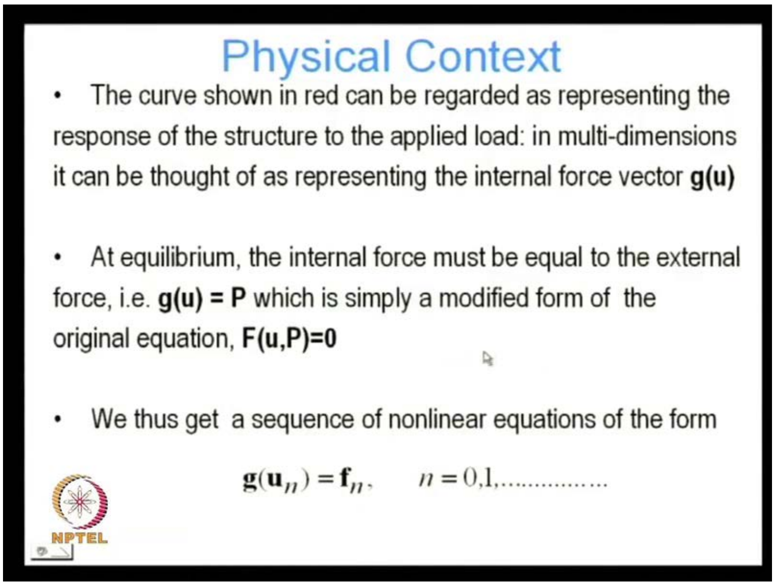
- For the purpose of understanding, it often helps if we consider a two-dimensional idealized schematic of the problem where the load vector **P** and the displacement vector **u** are treated as single scalars, purely for illustrative purposes.

It is satisfied for the purpose of understanding. It often helps if we consider a two-dimensional idealized problem, where the load vector and the displacement vector are treated as single scalars, but purely for illustrative purposes, just for understanding keep it mind that they are not really scalars. They are n dimensional vectors, but just for the sake of understanding, let us draw this in p use space instead of drawing my hyper surface in n dimensional hyper space which is going to be some strain surface which you cannot even visualize, right. So, if we draw, if we look at it, if we consider an idealization like this, you can see that this equilibrium solution is going to be all my points which satisfy equilibrium are going to lie on this red line, right. They are going to lie on this red line.

So, what do points on those red line mean? That means, at those points my u and p are such that they may force my external force again. This is becoming slightly more mechanics. So, we entered, but let us put it like this. Let us load. My applied load becomes equal to the system response, right. It is in equilibrium when my applied load becomes equal to my system response. What is my system response? My system response is a function of my displacements, right.

So, given a certain set of displacements, my system response through what is known as an internal force, it is an internal force which is the system response to my external excitation, external load and when is the structure going to be, when is it going to satisfy equilibrium, when the external response, external excitation or external load becomes equal to the internal response, right and we are going to represent the internal response by this little function g of u, right. You can think of g of u as the internal response of the system to my external load p and it is going to be in equilibrium when p is equal to g of u.

(Refer Slide Time: 52:21)



Physical Context

- The curve shown in red can be regarded as representing the response of the structure to the applied load: in multi-dimensions it can be thought of as representing the internal force vector **g(u)**

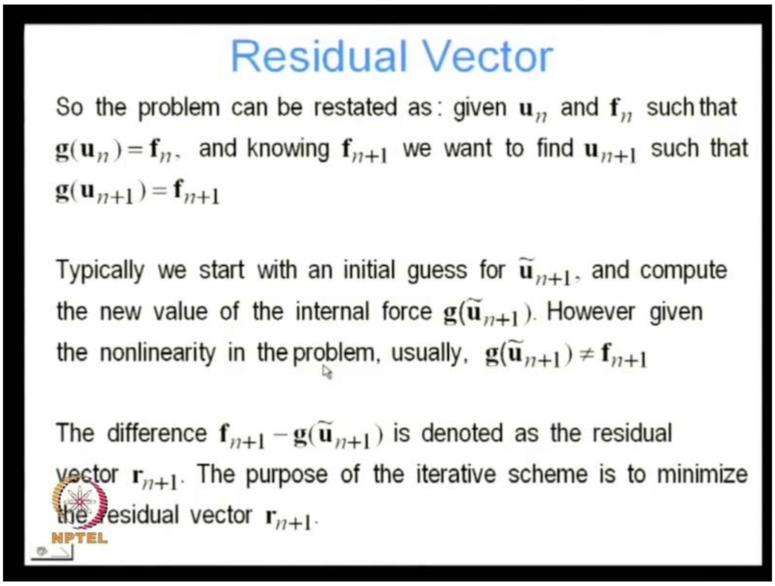
- At equilibrium, the internal force must be equal to the external force, i.e. **g(u) = P** which is simply a modified form of the original equation, **F(u,P)=0**

- We thus get a sequence of nonlinear equations of the form

$$\mathbf{g}(\mathbf{u}_n) = \mathbf{f}_n, \qquad n = 0,1,\ldots\ldots\ldots\ldots\ldots$$

So, the curve shown in red can be regarded as representing the response of the structure to the applied load in multi-dimensions. It can be thought of as representing the internal force vector g of u at equilibrium. The internal force must be equal to the external force, right. So, my internal response g of u must be equal to my external load p which is just writing this previous equation F of u p in slightly modified form. So, if I move p to the other side and whatever I left becomes g that is by g, so that I call my internal force and then this equation I have just re-written it to be that, right. So, instead of solving f of u p equal to 0, I am going to solve this equation iteratively, right. So, we are going to get a sequence of non-linear equations in this form. At each iterate, at each step I am going to solve this equation, right g of u n is equal to F of n where f of n is my external stimulus, u n is my system displacement, my independent variable and g of u n is my system response, right.

(Refer Slide Time: 53:45)



Residual Vector

So the problem can be restated as: given $\mathbf{u}_n$ and $\mathbf{f}_n$ such that $\mathbf{g}(\mathbf{u}_n) = \mathbf{f}_n$, and knowing $\mathbf{f}_{n+1}$ we want to find $\mathbf{u}_{n+1}$ such that $\mathbf{g}(\mathbf{u}_{n+1}) = \mathbf{f}_{n+1}$

Typically we start with an initial guess for $\tilde{\mathbf{u}}_{n+1}$, and compute the new value of the internal force $\mathbf{g}(\tilde{\mathbf{u}}_{n+1})$. However given the nonlinearity in the problem, usually, $\mathbf{g}(\tilde{\mathbf{u}}_{n+1}) \neq \mathbf{f}_{n+1}$

The difference $\mathbf{f}_{n+1} - \mathbf{g}(\tilde{\mathbf{u}}_{n+1})$ is denoted as the residual vector $\mathbf{r}_{n+1}$. The purpose of the iterative scheme is to minimize the residual vector $\mathbf{r}_{n+1}$.

So, the problem can be restated as given u n and f of n, such that g of u n is equal to f of n. So, we start with a state where I satisfy my equation. Remember that we start with a, right. So, we know that and since I am looking at load controlled problems, so suppose I know f n plus 1. So, I know the change in f, right. I change in my external load which has changed from f n to f n plus 1. I want to find u n plus 1 such that g of u n plus is equal to f n plus 1. So, we started with an initial equation f of u p is equal to 0, right and then we just change this to rewrite it in this form because that is easier to solve, right. Next time we are going to look at ways to solve this equation, right. We will look at Newton Raphson look at conditions, where Newton Raphson fails and then we will look at alternative algorithms which can be used under those situations.

Thank you.