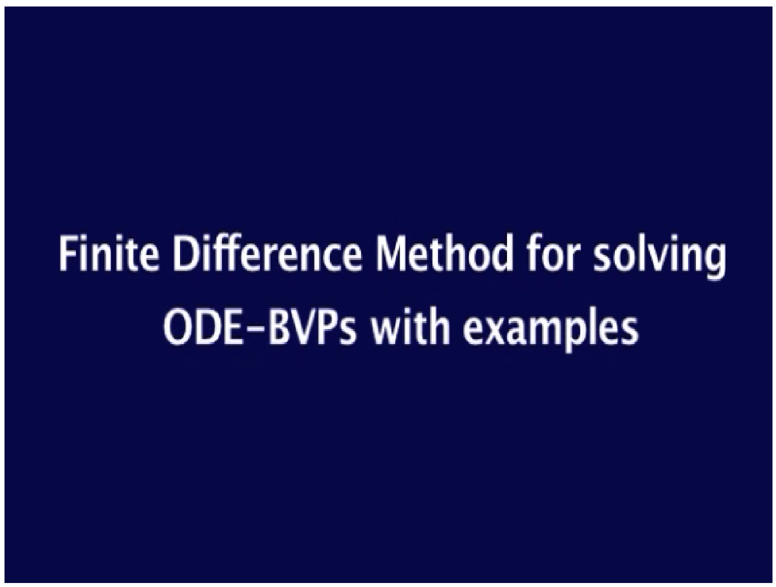
Advanced Numerical Analysis
Prof. Sachin Patwardhan
Department of Chemical Engineering
Indian Institute of Technology – Bombay

Lecture - 11
Taylor Series Approximation and Newton's Method

In the last lecture, we looked at the concept of a dense set and so we said that the set of rational numbers is dense on the real line and then any real number can be approximated using a rational number with an arbitrary degree of accuracy.

(Refer Slide Time: 00:43)

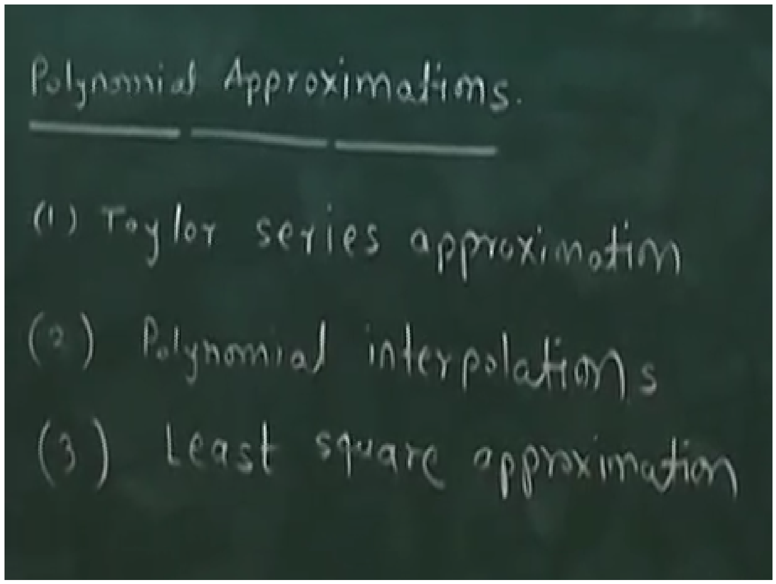


So that is one of the key result, which helps us in computing. We can represent a real number using the rational number and carry out approximate calculations, not exact calculations. In the same way, the set of polynomials is dense in the set of continuous functions over an interval A to B. This is a corner told result called Vieta's theorem and Vieta's theorem asserts that any function, any continuous function over an interval AB can be approximated by a polynomial with arbitrary degree of accuracy, a very, very important result.

Now as I explained in my last lecture, this is an existence result, it just guarantees that there exists a polynomial, which is a very a good approximation of given continuous function. It does not tell you how to construct this approximation. So it is an existent result, does not tell you how to construct approximations, but this basic idea that a continuous function can be approximated by a polynomial function forms the foundation of many, many, many of the numerical methods.

Where is it used for? It is used for transferring the problem into a computable form. So polynomial approximations are going be the core of next few lectures.

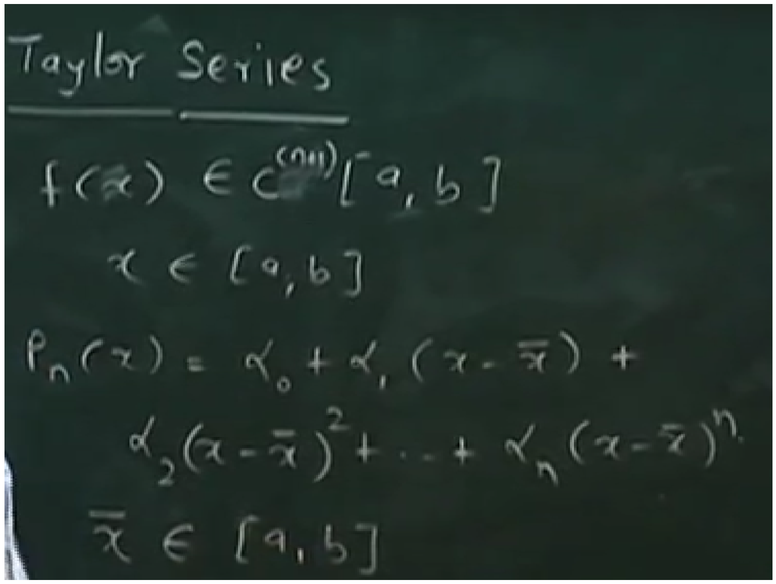
(Refer Slide Time: 02:22)



We are going to look at different ways of doing polynomial approximations. The first one is Taylor's series approximation. The second one, we will look at interpolation and the third one is least square approximations. By and laws we will stick to polynomial approximations, but we will also look at function approximations in between and so on. So these 3 basic ideas or these 3 basic tools give us a way of constructing these polynomial approximations.

So Vieta's theorem only gives you existence. Actually how do you construct these polynomial approximations will be done through these 3 approaches. Now let us begin with Taylor's series approximation.

(Refer Slide Time: 03:36)

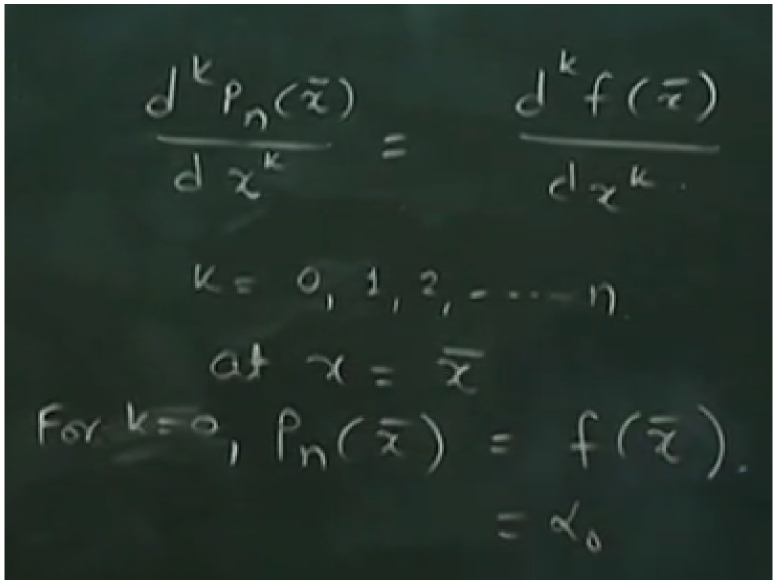


Let us begin our journey with Taylor's series approximations. Now Taylor's series approximation, if I am given a continuous function, say I have some Fx, which belongs to set of continuous functions over a, b and that means x is the independent variable, which varies between a and b. Now Taylor's series approximation allows me to construct a polynomial approximation with certain nice properties.

Now what is this nice property. Let us say this Pnx is the local approximation with alpha 0+alpha 1x-x bar+alpha 2x-x bar square up to alpha n, we cannot just look at the functions. When you are doing Taylor's series approximations, you cannot just talk about continuous functions, you need something more. You need differentiability. So we have to look at functions, which are not just once differentiable, which n+1 times differentiable.

Actually I have to work with a space, not C, I would work with Cn+1a, b, which means a set of functions, which are n+1*differentiable over interval a, b and x bar is some point that belongs to interval a, b. So neighborhood of a point x bar that belongs a, b, we want to construct a polynomial approximation, which is for f.

(Refer Slide Time: 06:00)



Now what is the characteristic of this approximation. The characteristic of this approximation is that the derivatives, the nice thing about this polynomial approximation is that derivatives of this polynomial are same as derivative of this function at x = x bar. At x=x bar, this polynomial approximation and the original function have identical derivatives. How many identical derivatives? n identical derivatives.

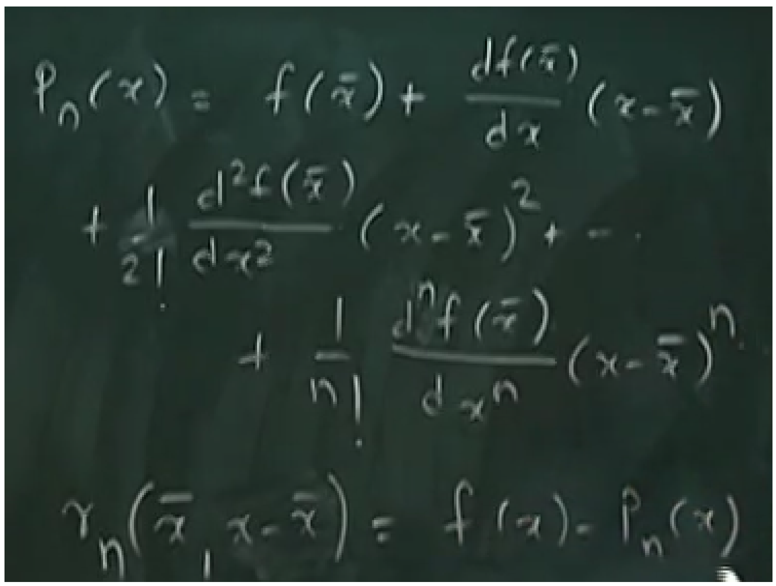
Now using this condition, it is very easy to derive what is alpha 0, alpha 1, alpha 2. If you start differentiating Pn, you will get different, so for 0-th order derivative, which means Pn x bar = f of x bar for K=0. So the first coefficient Pn x bar that is = alpha 0. So the first coefficient turns out to be this. The second coefficient, we just differentiate. So what is the second condition? The second condition is that dPn/dx at x=x bar should be = df/dx at x=x bar.

Notation that we have is f of x bar/dx. We are calling this as f of x bar/dx. So very easy to show that alpha 1=df x bar, this is just a notation saying that the derivative evaluated at x=x bar. So likewise I go on differentiating and it is very easy to show that. What will be the first derivative? The first derivative of this will be alpha 1+2 times and is equated to the derivative here, that will give you the corresponding term. Likewise, it is very easy to derive that alpha k=q/k factorial.

It is very easy to derive this, I leave the derivation to you. It is very, very easy. You just go on differentiating substitute x=x bar and equate right hand side = left hand side. You impose the condition that we have the derivatives of the polynomial approximation and derivatives of the original function should be identical. If you just impose that condition, very simple derivation to arrive at this general condition.

For this particular polynomial, any k-th coefficient is actually given by 1/k factorial dfk/dxk, so that is the k-th derivative of f at x=x bar. So I can express a given function in terms of a polynomial, whose coefficients are local derivatives. So this polynomial, it turns out to be, this Pnx, this turns out to be.

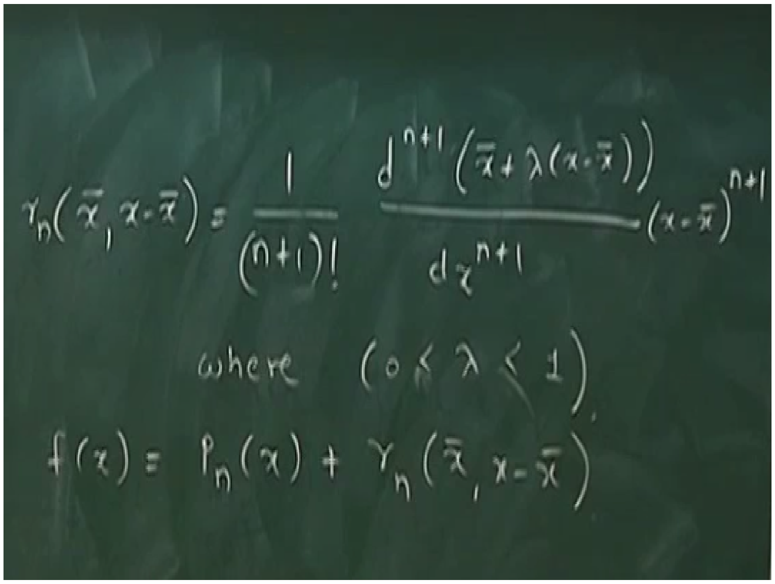
(Refer Slide Time: 10:37)



F at x bar +ds, so 1/2 factorial and so on, so 1/n factorial. So this you can prove very, very easily. Just equate a derivative, if you impose the condition that the approximation and derivative of the function at x=x bar should be identical, you will get this gap. Why should it be n+1 times differentiable? I have to now write f of x in terms of 2 components. So Taylor's theorem actually allows me to do 2 things.

One is it quantifies this polynomial, it allows me to construct this polynomial locally and then it also allows me to talk about the error, the residual. So we are now going to say that actually f of x. Let us define this Rn, let me define a residual, this Rn is a notation. Let me define the residual, which is function of x bar and x-x bar, which is difference between f of x, original function and the approximation. Now what Taylor's theorem tells you is.

(Refer Slide Time: 12:38)



So according to Taylor's theorem, Rn 1/n+1 factorial. So actually I can write f of x as Pnx+Enx. So this is exact expression. So this derivative n+1 derivative is not evaluated at x=x bar, it is evaluated at some intermediate point between x bar + lambda times delta x where lambda is a fractional value between 0-1. So actually derivation for this through Rolle's theorem an using mean value theorem, you can actually show that there exist some value of lambda.

For which you get exact equality and then that is how you actually prove this, but I am not interested right now in proving this. We are just going to use this result, so n+1-th derivative is required to define the residual term. This is an exact expression. You can remember that. This is an exact expression. So f of x exactly = this approximation + residual. The residual is defined by n+1-th derivative, that is why we need n+1 times differentiable functions.

Now this is something which you have studied in your undergraduate, we will see how we will be using this subsequently when it comes to finite difference method of solving OD value problems or partial differential equations. Now before I want to introduce a multivariable version of this. This is right now 1 scalar value x, what if you have a function in multiple variables, x1, x2, x3, x4 and xn. So what I am going to do now is to define a multivariable Taylor series.

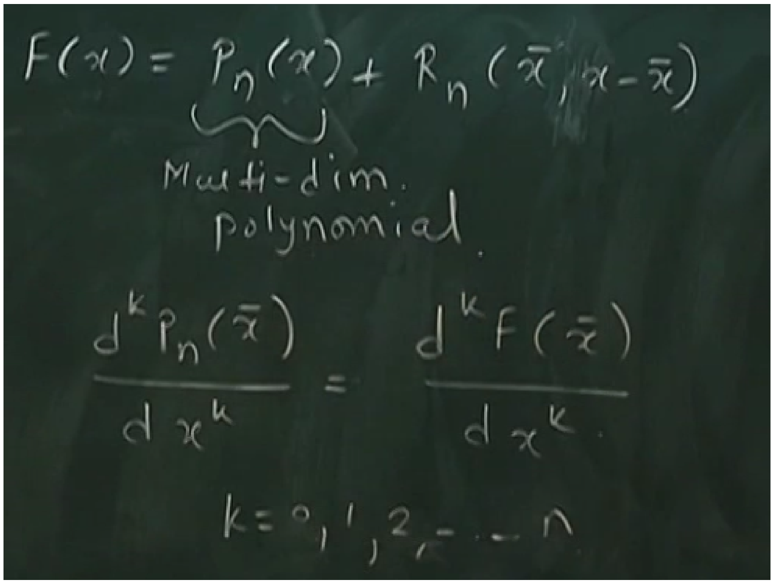
Which will conceptually be same, same concept is there that is you come up with an approximation whose higher derivatives are matched with the higher derivative of the function, same idea, except now we will start working with a function of a function vector in multiple variables and where is it used. I will also immediately derive one well known result for solving non-linear algebraic equations, that is Newton's method using the multivariable Taylor series.

So Taylor's series is not just defined for. Now let us say I have a function f of x which is from Rn to Rn. Here x belongs to Rn and I have a vector of functions. So f of x is actually, this f is a scalar function from n to 1, and then this is a vector of function. There is n such functions. Example, in the computing tutorial, we are looking at 4 equations related to a CSKR in 4 unknowns. Each one of them can be written as f1x, f2x, f3x, f4x.

There are 4 equations and 4 unknowns. In general, if you are trying to solve energy and material balance that is associated with some section of a chemical plant. You will get 1000 equations and 1000 unknowns. So you will have a vector of functions. Now can I extend the ideas of Taylor series approximation to these kind of function vectors. That is going to be critical for us in this course, where typically the use first at the most we use second derivatives.

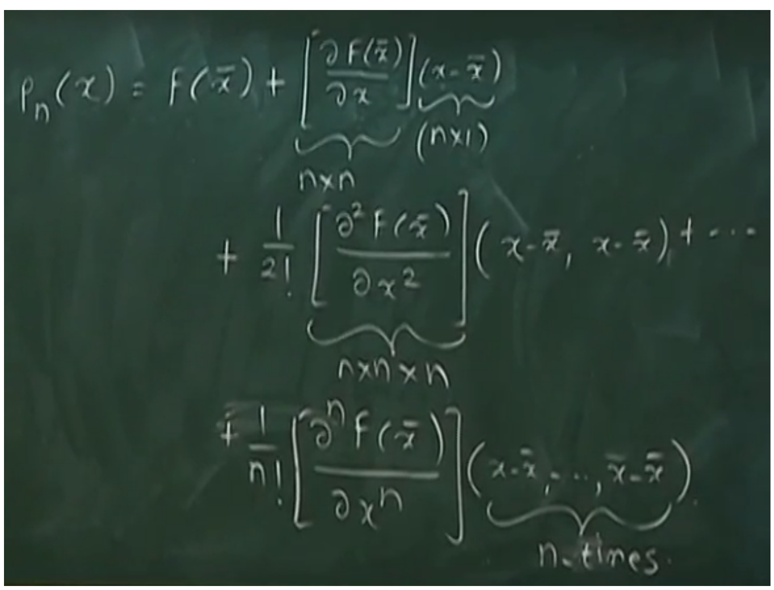
We do not really get into higher derivatives, but first and second derivatives of this function that they prove to be very useful in developing lot of methods. So what I am going to say now here is that just like the scalar case.

(Refer Slide Time: 17:50)



I can write f of x as Pnx+Rnx. Now this is a polynomial, but this is a multidimensional polynomial. This is not one variable polynomial. This is a multidimensional polynomial. How do you construct this? We still have the same condition, that is dkPn at x bar, so k-th derivative of this approximation should be matched with function derivative at that point. So this should be = dkf x bar, this is still the condition, for k=0, 1, 2, …n. What is my residual now?

(Refer Slide Time: 19:21)



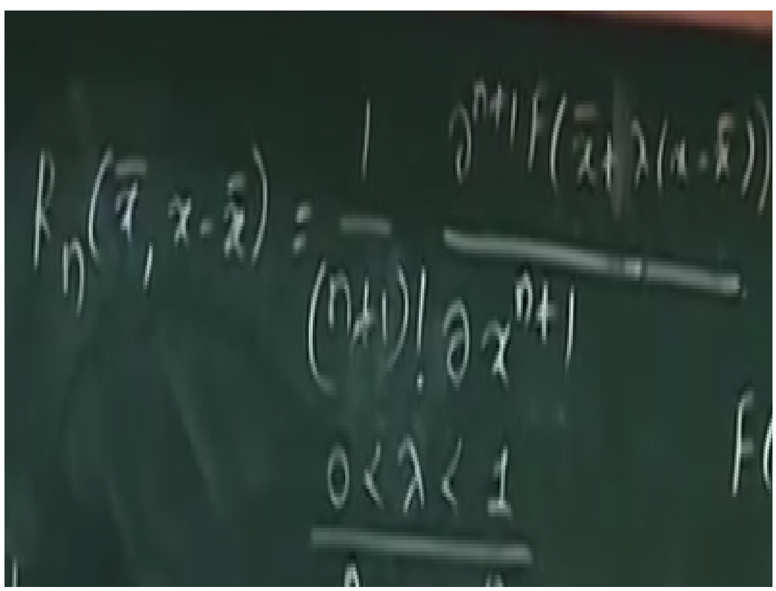
Let me first write down the function expansion. So Pnx, if I actually do a derivation here of matching the derivatives and then finding out the coefficients of Pn, I am just writing the final result because intermediate steps are very, very straight forward, would be f of x bar+, so this is the first coefficient of my polynomial, second coefficient is dou f/dou x, well computed at x bar. Now remember f of x is a vector. What will be dou f/dou x?

It will be a matrix; this will be a n x n matrix evaluated at a particular point. So this is a constant matrix. Once you evaluate it at a particular point, it is a constant matrix. This * x-x bar, this is a n x 1 vector. Then, the next term would be 1/2 factorial dou 2f/dou x square, but I am writing it in a little different way because this is a tensor. This is a tensor, this will be an n x n x n, when x-x bar operates on this once, you will get a matrix, when this operates twice, you will get a vector.

So this is sometimes in Maths is called bilinear matrix, but this is a tensor. This is an n x n x n couple. Then, you will have +1/n factorial dou n fx bar/dou x n. So this is a tensor, which is n x n x n, n times and then delta x operates on it n times to give me a vector. So actually in practice, in most of the numerical methods, we will be working with this first derivative, in some cases, in some situations, we may go to the second derivative, beyond that it becomes very, very difficult.

Nevertheless, the first derivative is very, very useful, as you will see that we will derive one very important method using this and what is the residual. The residual term here is; I will just write it down here.

(Refer Slide Time: 22:34)

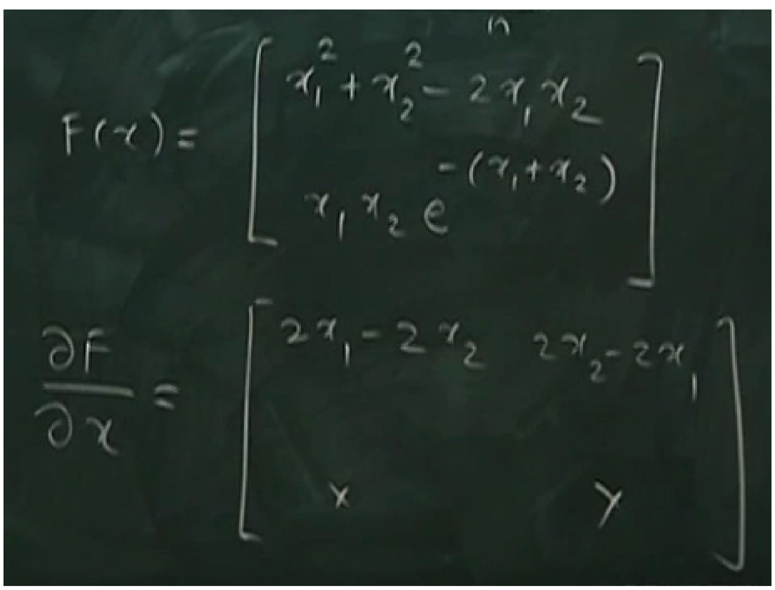


The residual here is 1/n+1 factorial. So this is n+1-th derivative evaluated at some point where x bar + lambda delta x, delta x is x-x bar and lambda is some value between 0-1. So this is exact expression. If you write polynomial approximation + the residual, together they form the exact expression. So this is for function vectors, which are n+1 times differentiable. Now function vector and its first derivative, Jacobian.

You have been calculating for the Newton Raphson method. The first thing I am going to do is to derive Newton Raphson method starting from this Taylor's series approximation. As an application of Taylor's series approximation, before I move into solving partial differential equations or boundary value problems, I want to show that this complex expression for n functions, written as a function vector is actually useful.

It is the practical application of this is going to be developing method for solving non-linear algebraic equations simultaneously. Is there any doubt, till now, anyone? This is just an extension. See because dou f/dou x, let us take an example. Let us take a simple example. I have given one example here.

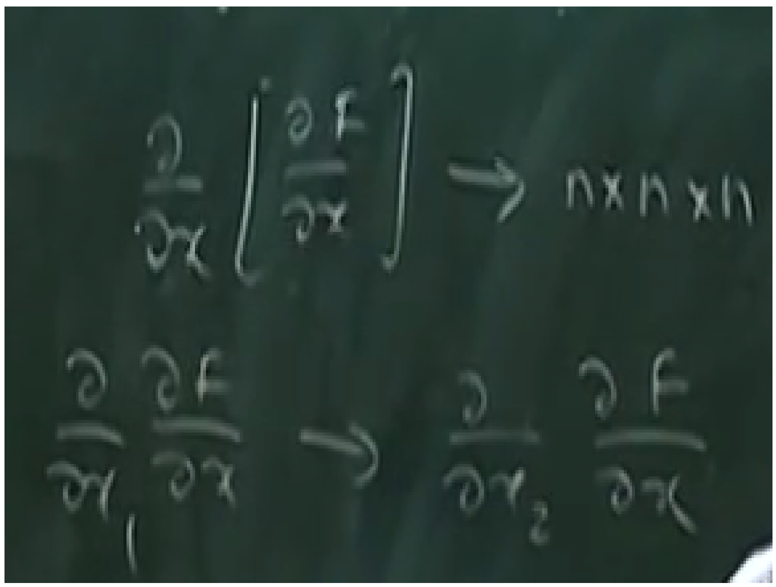
(Refer Slide Time: 24:47)



F of x is let us say x1 square + x2 square -2x1x2 and my second function is x1x2e to the power – x1+x2. This is my function letter. It is a function of 2 variables. So it is a function vector of 2 variables, this is the first function, this is the second function. What will be the Jacobian? What is dou f/dou x. So this will be 2x1-2x2, 2x2-2x1 and whatever here, will be 2 quantities here. Now what is the derivative of this. I will differentiate 4 entities.

I have differentiated now to come up with a higher derivative that his will be something, this will be what, this will be x2e to the power –x1+x2+something. There will some terms here, likewise. Now I want to differentiate this once more. What will I get? This is a matrix, matrix differentiated with respect to the vector.

(Refer Slide Time: 26:45)

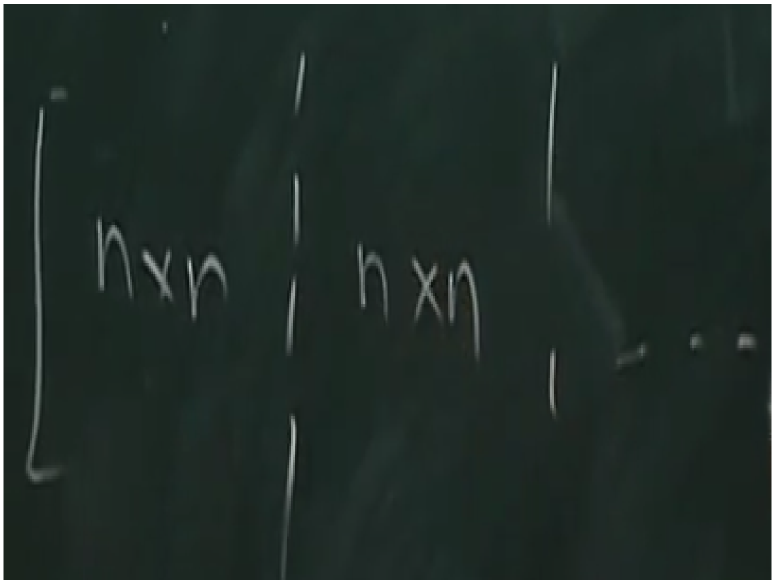


So dou/dou x of dou f/dou x, this will give me, this is n x n, this is a vector, which is n x 1, so I differentiate this I will get n x n x n. I mean simplest thing is dou f/dou x, so I can write dou f/dou x and dou/dou x1 of this will give me 1 matrix and then dou/dou x2 of dou f/dou x will give me another matrix. So it will be n x 2 x 2 and so on. So we go third derivative, it will be n x n x n x n. So it will just go up.

How do I operate? So there are rules of operating this, the way you differentiate, the different ways of writing this by linear matrix. So depending upon how you write it you can develop the rules for multiplication. So I can tell you a reference for where this is done, if you are interested, but during the course, we are not going to require the second derivatives, but you need them if you want to develop some advanced methods. So if you are interested, I can tell you references.

Basically what happens here is that this is a 3 dimensional array, once you operate x-x bar on it, you will get a matrix. You operate x-x bar on that matrix, you will get a vector. Because ultimately, you should get a vector here. This is the functional vector. So multiplication of this should give me a vector. You can decide some way of writing, I can write this as n x n matrix.

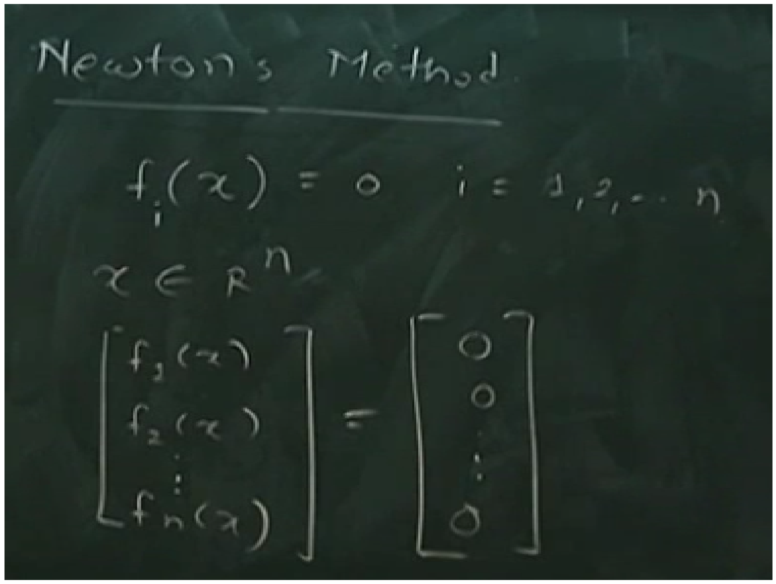
(Refer Slide Time: 28:55)



Then n x n matrix like this, actually it will not be, even if it looks like a matrix, it is not a matrix, because there are partitions, this is like a 3-D array. Up to 4D you can represent on paper, by somehow and any 4 dimensional, 5 dimensional in math-lab in computer, you can represent array of any dimension. Let us not worry too much about higher derivatives of function vector. What we are going to need most is the first derivative, that is Jacobian.

This Jacobian is most important for us in the course. So where is the application, where do I need this. So I a moving to section in terms of nodes, I move in to section 3.4, so I want to derive this Newton's method.

(Refer Slide Time: 29:54)



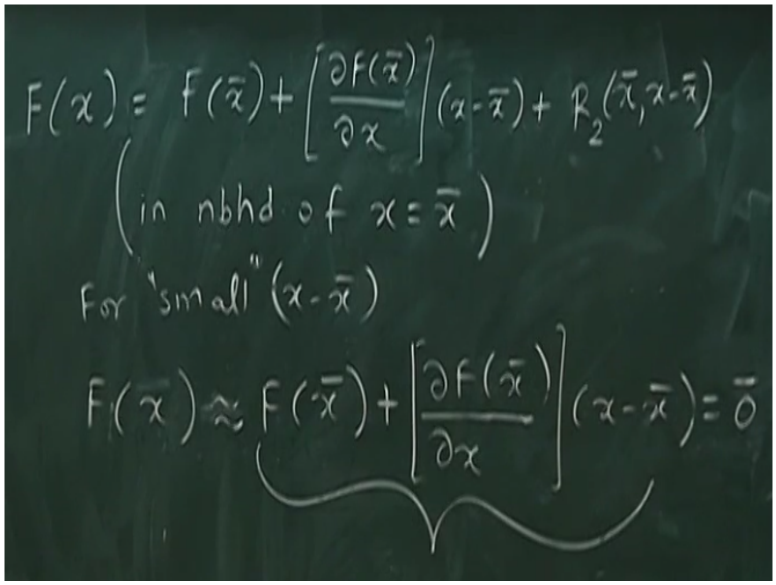
I want to derive Newton's method as an application of Taylor's series approximation. Now let us look at this problem, f of x, there are 2 functions, in this function vector. Let us say, I want to solve for this =0 and this =0. I want to simultaneously solve these 2 equations. You will get umpteen number of situations where you have to solve n non-linear equations in n variables simultaneously and get a solution.

Now if you have done the computing assignment, the first demo, you would have noticed that there are 2 equations. What do you mean by the 2 equations and so if I draw a graph of these equations in xy, x2 plane, we want to find out the points where these 2 graphs intersect. When it is line, if these 2 were linear equations, if these 2 were lines and in 2 dimensions, they meet only in 1 point, if at all they meet, or they could be parallel. There are 2 scenarios.

But for a non-linear equation, it is not like that. Non-linear equation, it could meet at multiple points. There could be multiple solutions for this particular problem. There is no unique solution when it comes to solving non-linear algebraic equations simultaneously. We want to develop a numerical method to reach a solution and I am going to use the idea of Taylor's series approximation to arrive at this method.

So my problem is that I want to solve for f of x, fix = 0, i=1, 2, ….n, simplest example as shown here. I want to solve for an x belongs to Rn or in other words, I want to solve for f1x, f2x = 0. I want to solve this problem. I want to solve n, non-linear algebraic equations, which are coupled. I want to solve them simultaneously. Now I am going to use Taylor's series approximation. Now what we know from Taylor's theorem for the multivariable case.

(Refer Slide Time: 32:57)



What I know from Taylor's theorem is f of x = fx bar. What I know from Taylor's theorem is, I can expand f of x in neighborhood of some x=x bar in the neighborhood of x=x bar, I can express it like this. If I am very, very close, if x is very close to x bar, I can actually ignore the higher order terms, I can ignore the second order and higher order terms for small. What is small here is n quotes. I am not going to precisely define what is small here.

If x-x bar is small, I can write fx=fx bar +dou f/dou x at x= x bar *x-x bar. Everyone get me on this. This is an approximation; this is not equal to. I am saying that this f of x for small, x-x bar is small, I can ignore the higher order terms in the polynomial expansion and I can say that f of x is almost = this, for small perturbation around x bar, x bar is some point. What is that I wanted to solve. I wanted to solve, let me go back here.

I wanted to solve this=0 or in a vector notation, which is nothing but f of x =0, I want to solve f of x =0, I am not able to solve this analytically f of x=0. I am trying to come up with some way of doing it iteratively. So I want to solve this, but I am not able to solve this, so I have approximated my original problem, I was talking about problem approximation, discretization. I use Taylor's series approximation and instead of solving for f of x=0, which is the original problem, I solve this =0.

Is this solvable? Why this is solvable, because this second derivative is calculated at x=x bar. So this is a matrix, which is once you calculate it at 1 particular point, this is a fixed matrix. What is this? Function vector evaluated at x=x bar, so this is n x 1 vector, this is a matrix, which gets fixed once you evaluate it at x=x bar, then this approximation is a linear equation, it is no longer a non-linear equation, approximation can be solved very easily.

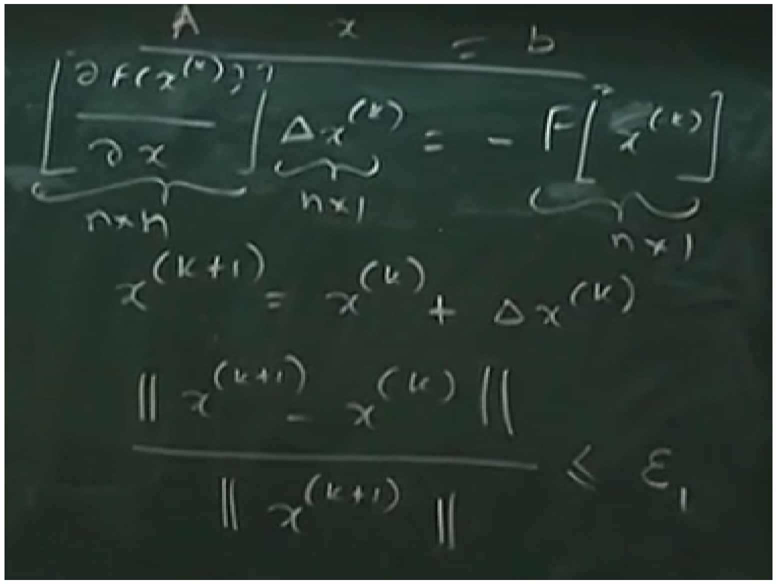
If I decide to solve this equation in place of my original equation, I decide to solve this equation. So what happens, I get a solution, x-x bar =, this is just a linear equation, -dou f/dou x bar, let me write in a live way, inverse of this matrix* f of x – f of x bar, this is a n x 1 vector. This is a n x n matrix, this problem is easily solvable and I get a new point x=x bar, so let me call this quantity as delta x, then I can write x=x bar + delta x.

If it will really happen that this new x, which you get here, you started from x bar, what you have done is like this, let us try to understand. I took a point x bar, let us say this my guess solution. I do not know what is the exact solution. I am guessing a solution; I am calling it x bar. Hopefully, this is close to the true solution. I should give a good guess, when I give a guess here. So my x bar is a good guess.

So around x bar, I linearized my original equation. I approximated as a linear equation or a first order polynomial in n dimensions to be very precise. We have ignored square terms, cubic terms, all n-th order terms, we just concentrated on the first derivative. Instead of solving the original problem, we solved this simplified problem and then this gave me a possible solution x, which is this. So I used this idea to come up with iterative scheme, which is Newton's method or sometimes called Newton Raphson method.

So Newton's method is basically now these 2 steps. How do I use it to come up with a iteration scheme? Let my x0 denote initial guess solution and then I am going to use Newton's step to come up with iteration scheme, which is like this.

(Refer Slide Time: 39:55)



I will call this as delta xk=-dou f/dou x inverse f of xk, xk+1=xk+delta xk. My raw Newton scheme is simply correction, how was the correction obtained? Using linearization in the neighborhood of which point, the previous point, I start with the guess x0, I linearized my non-linear equations locally, solve the linearized problem, get delta xk, and then this delta xk is used to create a new guess. From x0, I will get x1.

See x0+delta x0 will give me x1. Then I use x1, do the same thing, I get x2, I get x3, I get x4, so I get a sequence of vectors. So there are multiple things being discussed here. First, original problem is being approximated or discretized using Taylor's series approximation. We are not able to solve the original problem exactly. We are simplifying and solving the simplified problem. What we know very well is how to do Ax=B.

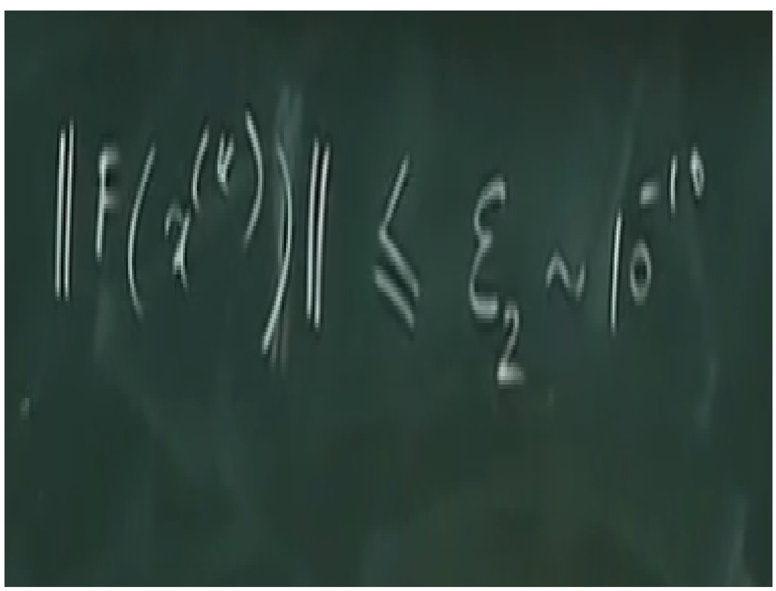
Actually this is nothing but Ax=B. Actually maybe I should write this not as an inverse, I should write this problem as dou f/dou xk delta xk=-f. This is an n x n matrix. This is a n x 1 vector. This is nothing but Ax=B. in abstract form, this is nothing but Ax=B. Solving linear algebraic equation, something which we know very well. We know how to solve linear algebraic equation, so I am solving Ax=B and then the delta x, which I get here is added to xk to create a new guess and then I continue this. This is very, very important.

If you want to get good conversions, initial guess is very, very important. That is where my input as an engineer, or a physicist, or a scientist will come into picture. I should know what values. I mean if I have a pressure or concentration or say mole fraction between 0-1, I cannot give a guess point 1.5 or -0.5, so physics comes there. Initial guess is very, very important. If there are multiple solutions, it may happen that if you give guess close to one solution, iterations will go to that solution.

If you give guess close to another solution, iterations will go to that solution. Now the question is, is this sequence quasi? Is this sequence converging? What do I do here is I look at xk+1-xk, norm of this, it is many times because of numerical problems, it is many times risky to look at only this difference, we should normalize it, so it is good to look at this for normalization. I would want to know whether this is <= epsilon.

If this is <= epsilon, I terminate my iterations, otherwise I just keep doing this. I start with an x not, initial guess, I keep doing these steps, so original problem, which is solving non-linear algebraic equations is converted into sequence of linear algebraic equations. I am solving linear algebraic equations again and again, hoping that this sequence will lead to solution of the original problem, so we need to check whether we are going there.

(Refer Slide Time: 44:51)



So we need 1 more convergence criteria and that is we need to check whether norm, if this is less than epsilon 2, ultimately why am I doing all this, I want to solve f of x =0, will I ever get 0 exactly, in the computer, I will never get 0. I have to give some epsilon to be some small value, this would be you know like 10 to the power -10 or something, some very, very small value, I want to say that do these iterations till any 1 of these conditions is satisfied.

That means doing more iterations is not helping me. I am just at the same point. If this goes close to 0, then I am doing iterations and I am at the same point. The same thing is here. If I am doing iterations and if this has gone, becomes sufficiently small, I can stop. So this is Newton's method which is developed by using multidimensional polynomial approximation, what kind of approximation, Taylor's series approximation.

This is something which we will be using again and again and you also have done programming for this. So you will get more inside into what is this. Now it will become clear, where is the sequence, why do you have to worry about, you know some sequence converging. When I start doing this iterations, I start from x0, I continue doing this till I get convergence of what, the sequence of vectors.

Each one of you, if I give you the same problem, each one of you might start from a different initial guess. Each one of you will get a different sequence of vectors. I have to worry about whether there is a convergence. Whether the sequence converges to a solution and so on. So what we have looked at is 1 application of Taylor's series. There are numerical difficulties. If you do not normalize, sometimes, your x is a vector which itself have very small quantities.

It may have more fractions, which are 10 to the power of -3, -4 and then the difference will look small, but actually it is not small. So you should look at relative error. Relative error is always better than, so that is likely to look at relative error. Next class, we will start looking at other applications of Taylor's series approximations in solving problems.