MATLAB Programming for Numerical Computation

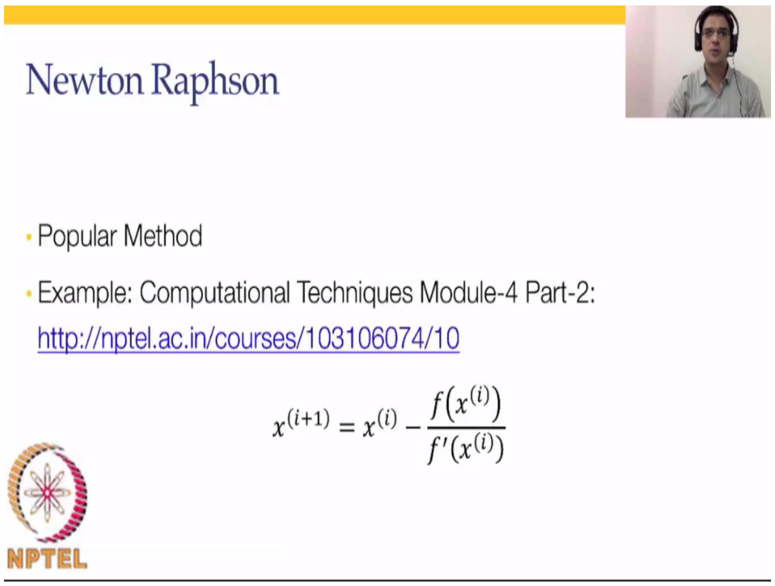
Dr. Niket Kaisare
Department of Chemical Engineering
Indian Institute of Technology, Madras

Module No. #05
Lecture No. #5.4
Non Linear Algebraic Equations – Newton Raphson (Single Variable)

Hello and welcome to MATLAB programming for numerical computations. In module 5 we have been covering methods to solve non linear algebraic equations of the form f (x) =0. In lecture 5.1 we covered a numerical technique known as bisection rule. Bisection rule we start with 2 initial guesses and use the bisection rule in order to obtain the desire solution. In lecture 5.3 we covered different numerical technique known as fix point iteration.

Fix point iteration uses only a single initial guess then we recursively solved equation in order to get the desired solution. In this lecture we are going to cover a third method known as Newton Raphson method which is a very popular method. Today we are going to cover a single variable version of Newton Raphson method. In a subsequent lecture we are going to extend the Newton Raphson method to multiple variable cases. However today what we are going to do is 2 things.

First we will see how we can use Newton Raphson in MATLAB in order to solve a problem f (x) = 0. And later on we will see how quickly Newton Raphson method converges to a desired solutions okay. So talk about Newton Raphsons.
(Refer Slide Time: 01:33)



Newton Raphson

- Popular Method
- Example: Computational Techniques Module-4 Part-2: http://nptel.ac.in/courses/103106074/10

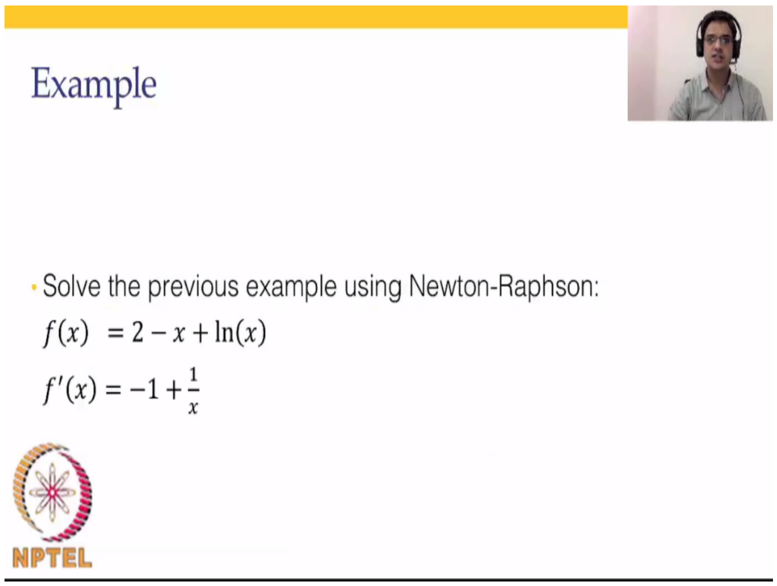
$$x^{(i+1)} = x^{(i)} - \frac{f\left(x^{(i)}\right)}{f'\left(x^{(i)}\right)}$$

NPTEL

It is a popular method which was covered again in module 4 part 2 of the computational techniques course the link for which given over here. Each new guess for Newton Raphson method we can start with an initial guess x0 and use this equation in order to obtain x1 and then we use x1 in order to obtain x2 so on and so forth.

We are interested in solving the problem f (x) = 0 and given the function f(x). We also need to calculate f`x that is the first derivative of the function okay. So if we have the value of f(x) and f`x at the current guess and then this is the equation that will be used in order to compute the new guess using the Newton Raphson techniques.

A Newton Raphsons technique is a popular technique because it has one of the fastest rates of convergence among the various methods use for solving non linear algebraic equations okay.
(Refer Slide Time: 02:35)



Example

- Solve the previous example using Newton-Raphson:

$$f(x) = 2 - x + \ln(x)$$

$$f'(x) = -1 + \frac{1}{x}$$

Let us look at now how we will use Newton Raphson in order to solve example that we have been covering so far in this module. The example is f (x) is given by 2-x + ln(x) and we need to find the solution to the particular non linear equations. As I said in the previous slide in order to calculate this solution using Newton Raphson method we also need f `(x).

And f`(x) for this problem is nothing but -1 + 1 / x okay. So once we are equipped with f (x) and f`(x), we can go over to MATLAB and use Newton Raphson to calculate the solution here using an iterative method okay. (Video Starts: 03:23) Let us open the file from the previous time which was fixed point iteration. And let us copy and save this as different file.

Let us call this as newtRaph, newtRaph okay. To solve non-linear equations using Newton Raphson there is no case1 and case2. Our f(x) is 2-x+ln(x) okay. So what I am doing over here in today's lecture is rather than starting from a blank slate, I will take the solution that we obtained in the previous lecture. And I will edit the solution in order to save the problem using Newton Raphson.

So this is one more way to doing this. We will able to write a Newton Raphson code little bit more quickly than if we were starting from a blank slate okay. We will again keep this same initial guess x0 =1. Let us keep maxIter as 50; tolX is just keep it as the power of -4 okay. This is where we need to change okay.

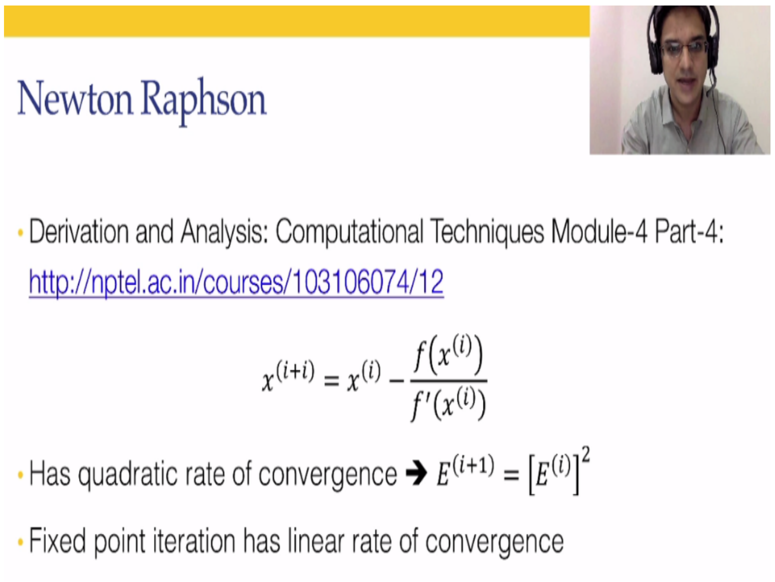
So for Newton Raphson method, first we need to calculate f (x) and f is nothing but 2-x+log (x) and f ‘, we will represent as df. df is -1+1/x right. So -1 comes from here and 1/x comes from here okay. So x i+1 is nothing but x-f / df. And that is our solution using Newton Raphson and then we continuing calculating the error we store x in xold.

And if error is less than the tolerance value, then we stop okay. So that is our Newton Raphson technique. Let us run it and let us look at the solution okay. So for an initial guess of 1 we actually do not get a solution and this is a demonstration of one of the problems with Newton Raphson. If we do not start with decent initial guess for Newton Raphson just like fixed point iteration.

We may not get a Newton Raphson to converge. Let say instead of 1 we were to start with different initial guess of 2. Let save this and run okay. If we were to start with an initial guess of 2, Newton Raphsons method converges in 4 iterations to the 3.1462. So let us head over back to power point and see what we have done so far. (Video Ends: 06:31) We took up the same code had used last time for fix point iteration and modified it using the Newton Raphson formula.

And this was the Newton Raphson formula. The new value of x is nothing but old value –f. Function f(x) / f'(x). So now that we have solve the problem using Newton Raphson. In MATLAB what we are going to do in the next few minutes is, to do some analysis of Newton Raphson technique. In fact we do analysis of both Newton Raphson and fix point iteration we have covered in lecture 5.3.

And we will compare how the 2 methods behave visually each other. We have already seen that fix point iteration method. Take something 10 iterations in order to converge. Whereas, Newton Raphson converges in something like 4 iterations.
(Refer Slide Time: 07:29)



Newton Raphson

- Derivation and Analysis: Computational Techniques Module-4 Part-4: http://nptel.ac.in/courses/103106074/12

$$x^{(i+i)} = x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})}$$

- Has quadratic rate of convergence ➔ $E^{(i+1)} = [E^{(i)}]^2$
- Fixed point iteration has linear rate of convergence

Why that is because Newton Raphson has what is known as the quadratic rate of convergence which basically means that the error i+1 iteration is approximately equal to error in ith iteration to the power 2, on the other hand fix point iteration has linear rate of convergence. For example what that means is, if let us say the error ith iteration for 0.1, then i+1 iteration will be the 10 ^ -2.

Then it will go 10 ^ -4, 10 ^ -8 so on and so forth. So it has usually a much faster rate of convergence compare to fixed point iteration. Derivation and analysis of this was covered in module part 4 of the computational techniques the link for which is given over here okay. So let us look at what at mean, when we talk about quadratic rate of convergence, so what we want to do is, want to look at how error in i+1th step compare with the previous step.

(Video Starts: 08:38) So let us go back to MATLAB and let us look at the Newton Raphson okay. So this was the overall Newton Raphson solution. So if we have defined our error in this particular manner, what I will do is, let us say I will call this error in ith step and so error i is this. So let me save this and let us run this and that we have 3 different error values okay.

So let us type this err vector okay. So this is error1, error2, error3, error4 okay. So prevErr is, prevErr is going to be error (1:3) and currErr = err (2:4) okay. So let me just do prevErr, to show what I am actually going okay. So prevErr was, so this is ei. This e1 and this is e2. This e2, this is e3, this is e4. If I put currErr as this y axis and prevErr as the x axis.

What we are plotting is e i+1 against e i in the x axis. So let us do that plot x axis is prevErr, prevErr y axis is currErr okay. Let us plot this okay. So this is how this currErr and the prevErr are going to change okay. So let us, what we also want to do is not have it in this way, we want have the logarithm of that.

So that we have some meaningful conclusion. So log (prevErr) and log (currErr) okay. So we have a log log plot and the slope of this lot is going to be approximately = 2. So we have this as x is 0.3 and y axis is-1. When x becomes -5. 6, y becomes -13. So -13, -1 is going to be a -12 and -5, -0.3, -5.6, -0.3 going to be approximately 6. So 12 / 6. So approximately speaking, slope of that line is a 3.

So we have this particular line as slope of 2 with a slope of 2 okay. So let us type hold on and that plot will be held let us go on to the other method, clear all, clc and that is fix point iteration. So added fix point iteration and let us do the same thing err (i). And we want to start with an initial guess of 2. So that is err (i) and we want to store the errors. Let run this, we are now store the errors okay.

What I need to do is this yeah, because I did not have err (i) < tol (x). What this was doing was, it if any err value is greater than tol (x) is not going to stop. So I made this particular change. Again clear all and let us run this okay. So in tenth iteration it has converged and what we will do is prevErr. PrevErr is 1: 9, currErr is 2 to 10 and what we want to do is plot loglog prevErr against currErr.

And let plot this as dash red line, show graph okay. So this is how the error changes when this is fixed point iteration and this is how the error changes when it is Newton Raphson. So in case of a fixed point iteration what we get is, a linear line x has fallen or log x has fallen by about 9, y has fallen so by approximately 9 from -1 to -10, y has gone. And from approximately 0 to -9, x has gone. So x verses y has a linear slope whereas, the slope is approximately the slope of 2.

So that it was mean the linear verses quadratic rate of convergence. (Video Ends: 14:23). When it is a quadratic rate of convergence error in the 1th step is square of the error in ith step and when it is linear rate of convergence ith step error is approximately the same order of magnitude as i-1th step error okay. So that is why partly Newton Raphson method popular because it has a quadratic rate of convergence.

What we have also seen in today’s lecture, when we gave a very small initial condition or Newton Raphson, when we started Newton Raphson with an initial condition of 1. Newton Raphson did not converge but it indeed diverges. So the divergence problem that we have actually seen for fixed point iteration problem is also there for Newton Raphson method. So one needs to be very careful before applying method such as Newton Raphson or fixed point iteration. So with that I come to the end of this lecture, lecture 5.4.

In this lecture and the previous lecture we have covered 2 techniques. The fixed point iteration in lecture 5.3 and in today’s lecture Newton Raphson method. We ended today’s lecture by doing a comparison between the convergence rate of Newton Raphson and fix point iteration. The aim of that was really to understand what it means by a quadratic rate of convergence and why the quadratic rate of convergence result in a popularity of Newton Raphson method.

As for as this particular lecture series goes, we are primarily interested in solving the problem using MATLAB and the error analysis for the Newton Raphson method is something that we just added on in order to give us flavor for how this techniques work. With that I come to the end of this lecture and I will see you in the next lecture. Thanks, and bye.