Lecture 19: Solution of System of Non-Linear Equations

(05-Spt-2012)

CE 601 NUM. MET.

LECTURE - 19
05- SPT- 2012

Solutions of Non-linear equations (Contd...)

In the last class, we discussed on:

* Application of Newton's method to find a simple root of polynomial $P_n(x)$
* Application of Newton's method to find a complex root of polynomial $P_n(x)$
* Newton's method to find roots that occur multiple times for the polynomial $P_n(x)$

$$x_{i+1} = x_i - \frac{m\, f(x_i)}{f'(x_i)}$$

or

$$x_{i+1} = x_i - \frac{f(x_i)\, f'(x_i)}{\left(f'(x_i)\right)^2 - f(x_i)\, f''(x_i)}$$

Today, first we will discuss on solution to non-linear systems.

System of Non-linear equations

→ Many problems in engineering and science involve non-linear equations

→ There may arise situations when you require to solve system of non-linear equations

ie
$$f(x_1, x_2, x_3, \ldots, x_n) = 0$$
$$g(x_1, x_2, x_3, \ldots, x_n) = 0$$
$$\vdots$$
$$h(x_1, x_2, x_3, \ldots, x_n) = 0$$

You have to find the solution $\{x\} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{Bmatrix}$

(Please Note that : → The suffix in x is not representing iteration number. They are suggesting the dimension).

→ You have a system of n- non-linear equation

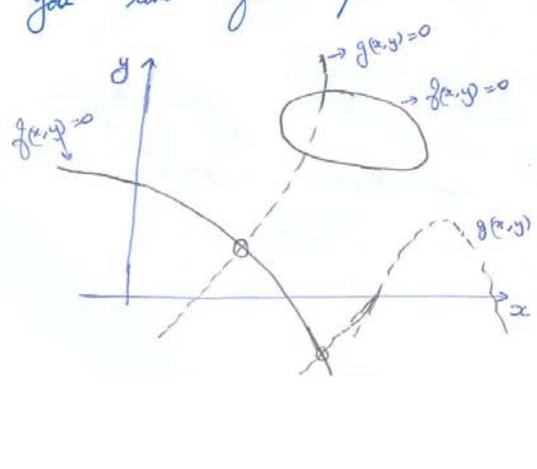
Consider for example a two - dimensional system

$$f(x, y) = 0$$
$$g(x, y) = 0$$

You want to find the solution $\begin{Bmatrix} x \\ y \end{Bmatrix}$

for the given system of two non-linear equations.

You can first plot the functions on an x-y plane.



→ Plot in the x-y plane the curves that have $f(x,y) = 0$ (or contours)

→ Similarly the contours having $g(x,y) = 0$ are drawn plotted.

The intersection points where you have $f(x,y)=0$ and $g(x,y)=0$ are solution to the given non-linear system.

⇒ As the term suggests due to non-linearity there are chances of more than one intersection point on the plane xy.

Q: How will you solve such non-linear systems?

Whatever methods you have seen for non-linear equations can be used for non-linear systems. Should be open domain method.

For simple cases you may use → Fixed Point Iteration

* Newton's method can be used in general for many types of non-linear systems
* Similarly Secant method, Muller's method, etc.

→ But how will you use these methods to solve a non-linear system?

As shown in the figure you have

$f(x,y) = 0$
$g(x,y) = 0$

→ Start with some initial guess (x_0, y_0)

→ Therefore at any iteration 'i', you have the improved pair (x_i, y_i)

→ Iterate till convergence.

Note:- At any iteration 'i' you have (x_i, y_i) as the known quantity.

Using Taylor's series:

$$f(x,y) = f(x_i, y_i) + \frac{\partial f}{\partial x}\Big|_{(x_i, y_i)}(x - x_i) + \frac{\partial f}{\partial y}\Big|_{(x_i, y_i)}(y - y_i) + \cdots$$

$$g(x,y) = g(x_i, y_i) + \frac{\partial g}{\partial x}\Big|_{(x_i, y_i)}(x - x_i) + \frac{\partial g}{\partial y}\Big|_{(x_i, y_i)}(y - y_i) + \cdots$$

Truncating after the first order term, you get

$$f(x,y) = f(x_i, y_i) + \frac{\partial f}{\partial x}\Big|_{(x_i, y_i)}(x - x_i) + \frac{\partial f}{\partial y}\Big|_{(x_i, y_i)}(y - y_i) = 0$$

$$g(x,y) = 0 = g(x_i, y_i) + \frac{\partial g}{\partial x}\Big|_{(x_i, y_i)}(x - x_i) + \frac{\partial g}{\partial y}\Big|_{(x_i, y_i)}(y - y_i)$$

or $$\frac{\partial f}{\partial x}\Big|_i (x - x_i) + \frac{\partial f}{\partial y}\Big|_i (y - y_i) = -f_i$$

$$\frac{\partial g}{\partial x}\Big|_i (x - x_i) + \frac{\partial g}{\partial y}\Big|_i (y - y_i) = -g_i$$

Please note I have changed the representation $f(x_i, y_i)$ as f_i, $\frac{\partial f}{\partial x}\Big|_{(x_i, y_i)}$ as $\frac{\partial f}{\partial x}\Big|_i$

$x - x_i \rightarrow$ The change required in x to
Now say Δx_i

$\text{IIIly}\ y - y_i \rightarrow \Delta y_i$

$$\therefore \left.\frac{\partial f}{\partial x}\right|_i \Delta x_i + \left.\frac{\partial f}{\partial y}\right|_i \Delta y_i = -f_i$$

$$\left.\frac{\partial g}{\partial x}\right|_i \Delta x_i + \left.\frac{\partial g}{\partial y}\right|_i \Delta y_i = -g_i$$

$$\begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix}_i \begin{Bmatrix} \Delta x \\ \Delta y \end{Bmatrix}_i = - \begin{Bmatrix} f \\ g \end{Bmatrix}_i$$

Solving this system will give the values of $\begin{Bmatrix} \Delta x \\ \Delta y \end{Bmatrix}$ in each iteration.

$$x_{i+1} = x_i + \Delta x_i$$

$$y_{i+1} = y_i + \Delta y_i$$

This is how you do iterative procedure.

An Example on Fixed-point Iteration

$$f(x,y) = x^2 - 2x - y + 0.5$$

$$g(x,y) = \cancel{x^2 + 4y^2 + 4y}\ x^2 + 4y^2 - 4$$

$\rightarrow$ To find solutions:

From $f(x,y) = 0 = x^2 - 2x - y + 0.5$

$$2x = x^2 - y + 0.5$$

or $$x = \frac{x^2 - y + 0.5}{2}$$

and from $g(x,y) = 0 = x^2 + 4y^2 - 4$

$$-8y = x^2 + 4y^2 - 8y - 4$$

or $$y = \frac{-x^2 - 4y^2 + 8y + 4}{8}$$

Assume some initial guess $(x, y)_{(0)}$

Then $$x_{i+1} = h_i = \frac{x_i^2 - y_i + 0.5}{2}$$

$$y_{i+1} = k_i = \frac{-x_i^2 - 4y_i^2 + 8y_i + 4}{8}$$

→ Do the iteration till convergence.

Newton-Raphson Method

For any n-dimensional non-linear system we can write the solution vector $\{x\} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{Bmatrix}$.

That is there are n-components for x.

→ Here onwards let me represent it as $\{x\} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} = \{x_j\}$;

$j = 1, 2, 3, \ldots, n$.

You require n - non-linear equations

$$f_1(x_1, x_2, \dots x_n) = 0$$
$$f_2(x_1, x_2, \dots x_n) = 0$$
$$f_3(x_1, x_2, \dots x_n) = 0$$
$$\vdots$$
$$f_n(x_1, x_2, \dots, x_n) = 0$$

(Please Note: → the suffixes here are not representing iteration numbers)
I will be incorporating iteration numbers in super fix in brackets
say $x_j^{(s+1)}$.

→ For the first equation

$f(x_1, x_2, \dots, x_n) = 0$ can be expressed in Taylor's series

Or let me say as stated earlier we want improvement $x_j^{(s+1)} = x_j^{(s)} + \Delta x_j$

$$f_1(\{x\} + \{\Delta x\}) = f_1(\{x\}) + \frac{\partial f_1}{\partial x_1}\Delta x_1 + \frac{\partial f_1}{\partial x_2}\Delta x_2 + \cdots \frac{\partial f_1}{\partial x_n}\Delta x_n = 0$$

i.e. $$\frac{\partial f_1}{\partial x_1}\Delta x_1 + \frac{\partial f_1}{\partial x_2}\Delta x_2 + \cdots + \frac{\partial f_1}{\partial x_n}\Delta x_n = -f_1$$

lly. $$\frac{\partial f_2}{\partial x_1}\Delta x_1 + \frac{\partial f_2}{\partial x_2}\Delta x_2 + \cdots + \frac{\partial f_2}{\partial x_n}\Delta x_n = -f_2$$
$$\vdots$$
$$\frac{\partial f_n}{\partial x_1}\Delta x_1 + \frac{\partial f_n}{\partial x_2}\Delta x_2 + \cdots + \frac{\partial f_n}{\partial x_n}\Delta x_n = -f_n$$

where you need to note that $f_1 \to f_1(x_1, x_2, \dots, x_n)$, etc.
$$\Delta x_1 = x_1^{(s+1)} - x_1^{(s)} \; ; \quad \Delta x_2 = x_2^{(s+1)} - x_2^{(s)}$$

i.e.
$$\begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & & & \\ \frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix} \begin{Bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{Bmatrix} = - \begin{Bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{Bmatrix}$$

$$[J] \{\Delta x_j\} = -\{g_j\}$$

or
$$\{\Delta x_j\} = -[J]^{-1}\{g_j\}$$

$$\Delta x_j^{(s+1)} = x_j^{(s+1)} - x_j^{(s)}$$

i.e.
$$\left\{x_j^{(s+1)} - x_j^{(s)}\right\} = -[J]^{-1}\{g_j\}$$

or
$$\left\{x_j^{(s+1)}\right\} = \{x_j\}^{(s)} - [J]^{-1}\{g_j\}$$

$[J] \rightarrow$ Jacobian

Note:- In modified N-R method the Jacobian matrix (J) is evaluated only once and in subsequent iterations the same $[J]$ is used.

→ By doing this you are reducing computational time to evaluate $[J]$ in each iteration.

As a Note: → Bairstow's Method

Coming back into finding solutions of non-linear polynomials $P_n(x)$, we have suggested how to linearly factorise $P_n(x)$

$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_1 x + a_0$

$P_n(x) = (x - \alpha)\, Q_{n-1}(x)$

→ You can also quadratically factorise $P_n(x)$

i.e. $P_n(x) = (x^2 - rx - s)\, Q_{n-2}(x) + \text{Remainder}$

for any value of 'r' and 's'.

→ For $(x^2 - rx - s)$ to be exact factor of $P_n(x)$, you require the remainder to be zero.

→ The ~~factored~~ polynomial is factored to $Q_{n-2}(x)$

$$Q_{n-2}(x) = b_n x^{n-2} + b_{n-1} x^{n-3} + \cdots + b_3 x + b_2$$

$$\therefore P_n(x) = (x^2 - rx - s)(b_n x^{n-2} + b_{n-1} x^{n-3} + \cdots + b_3 x + b_2) + b_1(x - r) + b_0$$

→ For $x^2 - rx - s$ to be exact factor of $P_n(x)$

you require $b_1(x - r) + b_0 = 0$

So that $b_1 = 0$, $b_0 = 0$

You have to select 'r' and 's' in such a way that $b_1 = b_1(r, s) = 0$ and $b_0 = b_0(r, s) = 0$

We get the coeffs $b_n, b_{n-1}, \dots b_3, b_2$ of $Q_{n-2}(x)$ as well as b_1, and b_0 using synthetic division.

Now Taylor's series expansion

$$b_1(r+\Delta r, s+\Delta s) \simeq b_1 + \frac{\partial b_1}{\partial r}\Delta r + \frac{\partial b_1}{\partial s}\Delta s + \cdots = 0$$

$$b_0(r+\Delta r, s+\Delta s) \simeq b_0 + \frac{\partial b_0}{\partial r}\Delta r + \frac{\partial b_0}{\partial s}\Delta s + \cdots = 0$$

i.e. $$\begin{bmatrix} \frac{\partial b_1}{\partial r} & \frac{\partial b_1}{\partial s} \\ \frac{\partial b_0}{\partial r} & \frac{\partial b_0}{\partial s} \end{bmatrix} \begin{Bmatrix} \Delta r \\ \Delta s \end{Bmatrix} = - \begin{Bmatrix} b_1 \\ b_0 \end{Bmatrix}$$

→ You use ~~synthetic~~ Newton's method to solve this system.

→ From synthetic division

$$b_n = a_n$$

$$b_j = a_j + r b_{j+1} + s b_{j+2}$$

$$j = n-1, n-2, \dots, 2, 1, 0.$$

→ Find a solution technique.

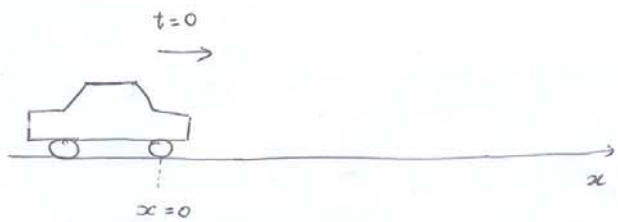
NEW CHAPTER (MODULE)

POLYNOMIAL APPROXIMATIONS AND INTERPOLATIONS

As engineering students, you may be required to handle lot of data.

- → Those data may be from experimental observation
- → Analysis, etc.

e.g. One can observe the distance moved by a car during certain time from its initial position. A person was observing how much distance the car moved at each 10.0 seconds.



Time, t (s)	Distance, x (m)
0.0	0.0
10.0	50.0
20.0	150.0
30.0	300.0
40.0	500.0
50.0	750.0
60.0	1000.0
70.0	1250.0
80.0	1500.0
90.0	1750.0

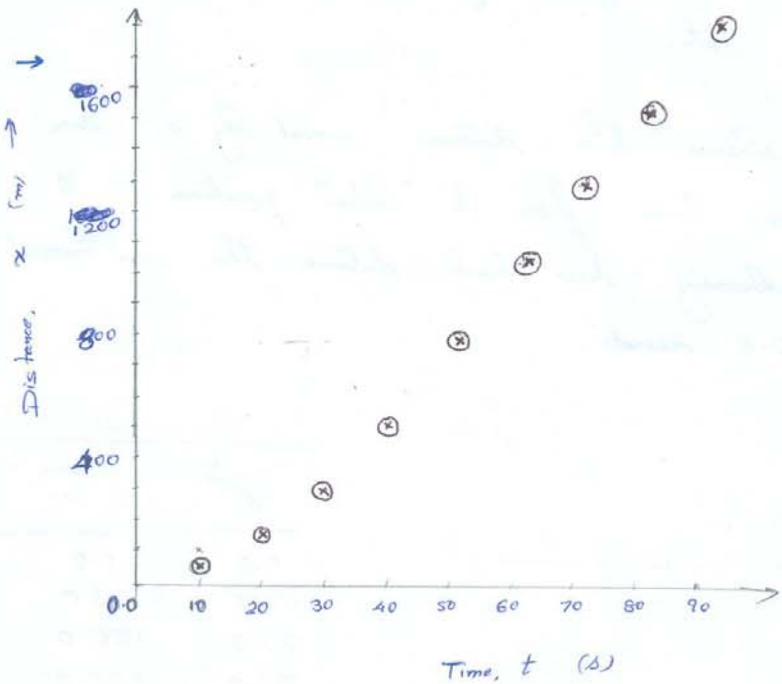
At time $t = 0.0$ seconds, we assume the distance $x = 0.0$ m.
Thereafter at every 10.0 seconds the distance x is measured.

- → The table above represents the experimental observation.
- → The time 't' and distance 'x' are observed as discrete values.

→ You can plot these observations as data points on an ~~set~~ x vs. t plane.

→ However, as the data is discrete, we will not be able to interpret distance moved at time say 15 seconds or 25 seconds, 38 seconds, etc. (See figure below).

→



→ ~~To~~ But being an engineer, you are interested to know the following properties of the car

- ★ Velocity
- ★ Acceleration

→ Also you will have interest how much distance it might have moved (say around 55 seconds or 75 seconds, etc.).

Method

→ Using these discrete data, one need to fit an approximate function using your numerical methods knowledge.

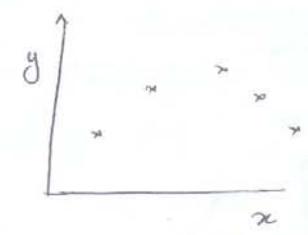
→ That is, there could be some exact function between time and distance (which we may not be aware). Instead of that exact function we may approximate using some of our known functions.

→ Then using these approximated functions you can infer the value of distance 'x' at any arbitrary time 't' in the range 0 - 90 seconds.

⇒ In the same way (as cited in the above example) there can be many type of observations one can plot say 'x' versus 'y'.

→ You need to approximate y as function of x in that case

i.e $y = f(x)$

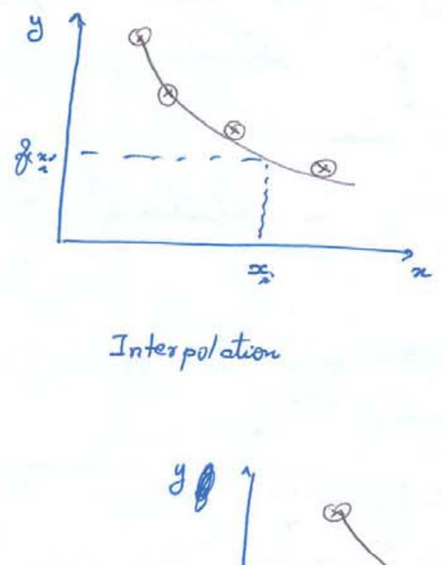


There are many types of approximating functions

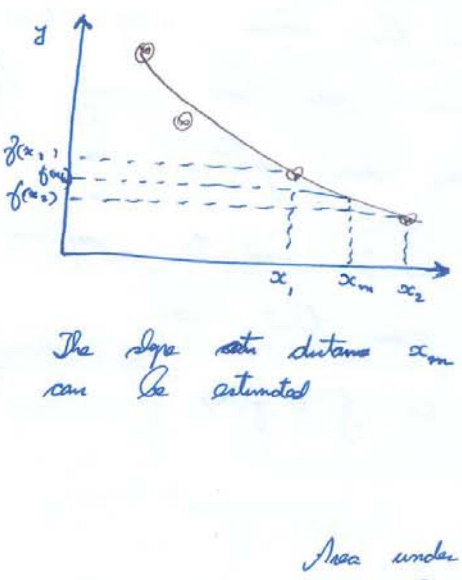
* Polynomials
* Trigonometric functions
* Exponential functions, etc.

The objective in deciding approximating function is:
- → It should be easy to evaluate
- → Easy to differentiate
- → Easy to integrate, etc.

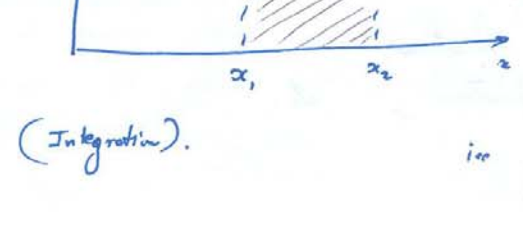
Using these approximating functions, one can
- → Interpolate the functional value for any inter-mediate point, or vice versa.
- → Find the slope of the curve generated by data point (or can aid in differentiation).
- → Or you can find area generated within certain data points



Interpolation



The slope at distance x_m can be estimated



(Integration).

Area under the curve between x_1 and x_2 can be obtained by integrating $f(x)$ i.e. $\int_{x_1}^{x_2} f(x)\,dx$.