The Newton-Raphson method for solving equations

By Gilberto E. Urroz, Ph.D., P.E.
January 2010

1 - The Newton-Raphson for solving single equations:

The Newton-Raphson method used for solving an equation of the form

$$f(x) = 0$$

requires the knowledge of the derivative f'(x). This can be easily accomplished in SMath Studio using the "Derivative" option in the "Functions" palette:

$$fp(x) = \frac{d}{dx} f(x)$$

Given an initial guess of the solution, $x = x_0$, the solution can be approximated by the iterative calculation:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

for $k = 0$,1, ...

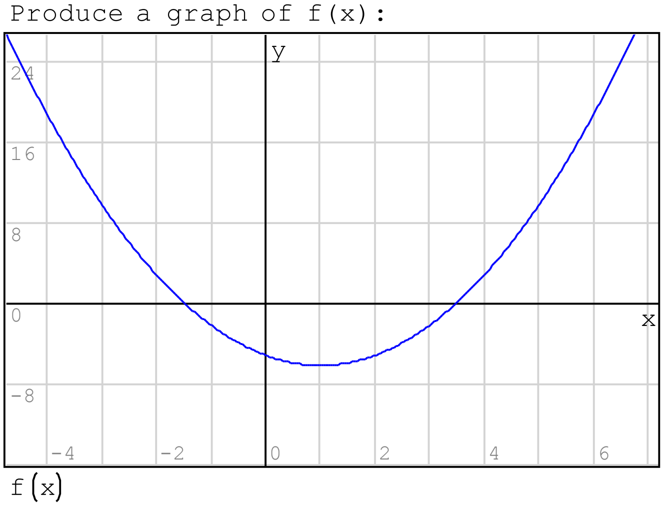
The iteration continues until either the solution converges, i.e., $\left|f(x_{k+1})\right| < \varepsilon$, or a certain large number of iterations are performed without convergence, i.e., $k > n_{max}$

Example: Solve the equation: $x^2 - 2 \cdot x - 5 = 0$

Solution: A graph of the function can help us find where the solutions may be located:

Define the function: $f(x) := x^2 - 2 \cdot x - 5$

Produce a graph of f(x):



The graph shows solutions near x = -2 and x = 3. We can implement the solution using the Newton-Raphson method as follows:

$$fp(x) := \frac{d}{dx} f(x) \qquad fp(x) \rightarrow 2 \cdot (-1 + x)$$

Parameters of the solution are: $\varepsilon := 1.0 \cdot 10^{-6}$ $nmax := 100$

First solution:

Starting with a guess of $xG := -2.5$

we find a solution by using the following iterative procedure:

$k := 0$

```
while ((k ≤ nmax) ∧ (|f(xG)| > ε))
    xGp1 := xG - f(xG)/fp(xG)
    k := k + 1
    xG := xGp1
```

$xG = -1.4495$ This is the solution found

$k = 4$ After this many iterations

$f(xG) = 2.1427 \cdot 10^{-11}$ The function at the solution point

Second solution:

Starting with a guess of $xG := 4.2$

we find a solution by using the following iterative procedure:

$k := 0$

```
while ((k ≤ nmax) ∧ (|f(xG)| > ε))
    xGp1 := xG - f(xG)/fp(xG)
    k := k + 1
    xG := xGp1
```

$xG = 3.4495$ This is the solution found

$k = 4$ After this many iterations

$f(xG) = 2.2529 \cdot 10^{-13}$ The function at the solution point

2 - Solution to equations with function "solve":

Most equations can be solved using function "solve" in SMath Studio. For the present case we'll have:

$$\text{solve}(f(x) = 0, x) = \begin{pmatrix} -1.4495 \\ 3.4495 \end{pmatrix}$$

Alternatively, you can use:

$$\text{solve}(x^2 - 2 \cdot x - 5 = 0, x) = \begin{pmatrix} -1.4495 \\ 3.4495 \end{pmatrix}$$

3 - The Newton-Raphson method for a system of equations:

A system of n equations in n unknowns can be represented as:

$$f1(x_1, x_2 .. x_n) = 0$$
$$f2(x_1, x_2 .. x_n) = 0$$
$$...$$
$$fn(x_1, x_2 .. x_n) = 0$$

or simply, $f(x) = 0$, with

$$f(x) = \begin{pmatrix} f1(x_1, x_2 .. x_n) \\ f2(x_1, x_2 .. x_n) \\ . \\ fn(x_1, x_2 .. x_n) \end{pmatrix} = \begin{pmatrix} f1(x) \\ f2(x) \\ . \\ fn(x) \end{pmatrix}$$

The variable x is defined as the vector:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ . \\ x_n \end{pmatrix}$$

We can provide an initial guess for the solution, x_0, and proceed with an iterative process defined by the formula:

$$x_{k+1} = x_k - J(x_k)^{-1} \cdot f(x_k)$$

for $k = 0$,1,... In this formula, $J(x_k)$, is the Jacobian matrix of the function defined as [to be 100% correct the derivatives in this matrix should be partial derivatives]:

$$J(x_k) = \begin{pmatrix} \frac{dy_1}{dx_1} & \frac{dy_1}{dx_2} & . & \frac{dy_1}{dx_n} \\ \frac{dy_2}{dx_1} & \frac{dy_2}{dx_2} & . & \frac{dy_2}{dx_n} \\ . & . & . & . \\ \frac{dy_n}{dx_1} & \frac{dy_n}{dx_2} & . & \frac{dy_n}{dx_n} \end{pmatrix}$$

Example:

How to calculate the Jacobian matrix of a system of three equations. Given the system of three equations:

$$f(x) := \begin{pmatrix} x_1 + x_2 + x_3 - 6 \\ x_1 \cdot x_2 \cdot x_3 - 6 \\ {x_1}^2 + {x_2}^2 + {x_3}^3 - 14 \end{pmatrix}$$

This is obvious, but could useful for larger functions:

$n := \mathrm{length}(f(x))$

$n = 3$

The following nested "for" loops calculate the elements of the jacobian matrix as the elements "jac[i,j]":

$$\begin{aligned} &\text{for } i \in 1..n \\ &\quad \text{for } j \in 1..n \\ &\qquad jac_{i\,j} := \frac{d}{d\,x_j} f(x)_i \end{aligned}$$

The following definition creates the function "Jacobi" that represents the Jacobian matrix of the function f(x) shown earlier:

$\mathrm{Jacobi}(x) := jac$

$$\mathrm{Jacobi}(x) \rightarrow \begin{pmatrix} 1 & 1 & 1 \\ x_2 \cdot x_3 & x_1 \cdot x_3 & x_1 \cdot x_2 \\ 2 \cdot x_1 & 2 \cdot x_2 & 3 \cdot {x_3}^2 \end{pmatrix}$$

Note: This approach for calculating the Jacobian matrix of a vector function was made available by Radovan Omorjan (omorr) in the SMath Studio wiki page: http://smath.info/wiki/diff.ashx

----

Generalized Newton-Raphson method for solving a system of equations:

The parameters of the solution are: $nmax := 100$ $\varepsilon := 1 \cdot 10^{-20}$

An initial guess is: $xG := \begin{pmatrix} 5 \\ -1 \\ 4 \end{pmatrix}$

The iterative process for the solution is expressed as:

$k := 0$

$$\begin{aligned} &\text{while } \big((k \le nmax) \wedge (\max(f(xG)) > \varepsilon)\big) \\ &\quad xGp1 := xG - \mathrm{Jacobi}(xG)^{-1} \cdot f(xG) \\ &\quad k := k + 1 \\ &\quad xG := xGp1 \end{aligned}$$

A solution is found after these many iterations:

$k = 15$

Here's a solution: And the function at that point:

$xG = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$

$f(xG) = \begin{pmatrix} -1.0161 \cdot 10^{-14} \\ -3.9933 \cdot 10^{-14} \\ -2.1316 \cdot 10^{-14} \end{pmatrix}$

Note: The function representing the system of equations solved above, namely,

$$f(x) := \begin{pmatrix} x_1 + x_2 + x_3 - 6 \\ x_1 \cdot x_2 \cdot x_3 - 6 \\ x_1^{\,2} + x_2^{\,2} + x_3^{\,3} - 14 \end{pmatrix}$$

can be thought of representing the system of equations:

$$\begin{array}{lcl} x + y + z - 6 = 0 & & x + y + z = 6 \\ x \cdot y \cdot z - 6 = 0 & \text{or} & x \cdot y \cdot z = 6 \\ x^2 + y^2 + z^2 - 14 = 0 & & x^2 + y^2 + z^2 = 14 \end{array}$$

with the variable substitution: $x_1 = x$, $x_2 = y$, and $x_3 = z$.
