Root Estimation using

Newton-Raphson

by

Hammond Mason

1. Univariate Case

Consider the Taylor Series of a function, f, of one variable, x:

$$f(x + dx) = f(x) + \frac{\partial f}{\partial x}.dx + \frac{1}{2}.\frac{\partial^2 f}{\partial x^2}.(dx)^2 + \cdots$$

As an approximation:

$$f(x + dx) \approx f(x) + \frac{\partial f}{\partial x}.dx$$

We can use this relation to find the value of dx such that $f(x + dx) \approx 0$ ie. the value $x + dx$ will be close to the 'root' of f.

If $f(x + dx) \approx 0$ then:

$$0 \approx f(x) + \frac{\partial f}{\partial x}.dx$$

$$\therefore -f(x) \approx \frac{\partial f}{\partial x}.dx$$

$$\therefore -\frac{f(x)}{\frac{\partial f}{\partial x}} \approx dx$$

$$\therefore dx \approx -\frac{f(x)}{\frac{\partial f}{\partial x}}$$

So, we can use this value of dx to find $x + dx$, ie:

$$x + dx \approx x - \frac{f(x)}{\frac{\partial f}{\partial x}}$$

This new value of x (x_{new}):

$$x_{new} = x + dx \approx x - \frac{f(x)}{\frac{\partial f}{\partial x}}$$

gives rise to a new value of f (f_{new}):

$$f_{new} = f(x_{new}) = f(x + dx) \approx f\left(x - \frac{f(x)}{\frac{\partial f}{\partial x}}\right)$$

which will be closer to 0 than $f(x)$ was. Of course if the value of f_{new} is not as close to 0 as we would like, we can just set the value of x to be x_{new} and perform the process again, ie.:

$$x_{new} = x_{old} - \frac{f(x_{old})}{f'(x_{old})}$$

where $f'(x_{old}) = \left.\frac{\partial f}{\partial x}\right|_{x=x_{old}}$

We can keep on iterating like this until our value for f_{new} is "close enough" to 0, ie. within our desired tolerance. This is the essence of the Newton-Raphson algorithm for finding the root of a function.

In applying Newton-Raphson there is one practical problem: determining $f'(x)$. Quite often the functional form of f is not known, eg. $f(x)$ could simply be the value in a cell of a complex spreadsheet. Not knowing the equation for $f(x)$ means not being able to determine $f'(x)$. In such a situation, however, $f'(x)$ can be approximated using the difference operator:

$$f'(x) \approx \frac{f(x + dx) - f(x)}{dx}$$

The smaller dx is, the closer the difference operator will come to the true value for $f'(x)$.

The Newton-Raphson algorithm, therefore, can be stated as:

1. Start with an initial value for x, being a guess as to what the root of the function is;
2. Set $x_{old} = x$;
3. Calculate $f_{old} = f(x_{old})$;
4. Compare f_{old} with 0 to see how far away from 0 we are;
5. If f_{old} is close enough to 0 then stop – we have found that x_{old} is close enough to the real root of f.
6. If f_{old} is not close enough to 0 then calculate $f'(x)$. This requires two separate calculations: one to calculate $f(x + h)$ and the other to calculate $f'(x)$;
7. Calculate x_{new} from the values of $f'(x)$, f_{old} and x_{old} we have just determined above, using the formula specified above;
8. Set $x_{old} = x_{new}$;
9. Go to step 3 with this new value of x_{old}.

2. Multivariate Case

The above univariate case can be extended to the multivariate case quite simply. Consider the two variable case: $f(x_1, x_2)$. Again, we start with the Taylor Series expansion of f, but this time with two variables, x_1 and x_2:

$$f(x_1 + dx_1, x_2 + dx_2) = f(x_1, x_2) + \frac{\partial f}{\partial x_1}.dx_1 + \frac{\partial f}{\partial x_2}.dx_2 + \cdots$$

Again, as an approximation:

$$f(x_1 + dx_1, x_2 + dx_2) \approx f(x_1, x_2) + \frac{\partial f}{\partial x_1}.dx_1 + \frac{\partial f}{\partial x_2}.dx_2$$

If $f(x_1 + dx_1, x_2 + dx_2) \approx 0$ then:

$$0 \approx f(x_1, x_2) + \frac{\partial f}{\partial x_1}.dx_1 + \frac{\partial f}{\partial x_2}.dx_2$$

$$\therefore -f(x_1, x_2) \approx \frac{\partial f}{\partial x_1}.dx_1 + \frac{\partial f}{\partial x_2}.dx_2$$

In matrix notation:

$$\therefore -f(x_1, x_2) \approx \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} . \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix}$$

The problem here, of course, is that we have one equation but two unknowns resulting in an infinite number of solutions. Therefore, in order to have a unique solution we require two equations, ie. two different functions which both depend on x_1 and x_2. If this is the case then we have the equations:

$$-f_1(x_1, x_2) \approx \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \end{bmatrix} . \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix}$$
$$-f_2(x_1, x_2) \approx \begin{bmatrix} \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} . \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix}$$

which simplifies to:

$$-\begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} \approx \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} . \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix}$$

or

$$-\mathbf{f} \approx \mathbf{J}.\mathbf{dx}$$

where $\mathbf{f}$ is the vector of $f_i(\mathbf{x})$, $\mathbf{J}$ is the Jacobian matrix and $\mathbf{dx}$ is the vector of x_is. Rearranging for $\mathbf{dx}$ gives:

$$\mathbf{dx} \approx -\mathbf{J}^{-1}\mathbf{f}$$

and the resulting vector from

$$\mathbf{x} - \mathbf{J}^{-1}\mathbf{f}$$

provides our estimates of $x_i + dx_i$ which we can use to generate $\mathbf{f}_{new}$ which can then be tested to see if each f_i is close enough to the target value to make $\mathbf{x} - \mathbf{J}^{-1}\mathbf{f}$ our estimate of the root of $\mathbf{f}$.

The problem with multivariate Newton-Raphson is that the more variates there are, the more calculations need to be done. When we had one variate, we needed to calculate the estimate of $f'(x)$ which involved 2 calculations: $f(x+dx)$ and $f'(x)$. With the bivariate case presented above we now have to calculate 4 partial derivatives, with each partial derivative requiring 2 calculations, ie. a total of 8 calculations to get $\mathbf{J}$. A trivariate case has 9 partial derivatives in its Jacobian matrix, ie. 18 calculations for $\mathbf{J}$. Generalizing we see that for the n-variate case, the calculation of $\mathbf{J}$ requires $2n^2$ separate calculations to be done. And this is before we even start to invert $\mathbf{J}$. Consequently, while Newton-Raphson lends itself well to the multivariate case, in its application we must be mindful of the computational effort required.