Newton's method for systems of non-linear equations

Bård Skaflestad

October 3, 2006

Multi-variate Taylor expansions

We wish to develop a generalisation of Taylor series expansions for multi-variate real functions, i.e. real functions of more than one real variable.

The bi-variate case

To this end, let $f : \mathbb{R}^2 \to \mathbb{R}$ be a sufficiently differentiable function of two real variables. We define the auxillary single-variable functions $u_y(x)$ and $v_x(y)$ by the relations

$$\begin{aligned} u_y(x) &= f(x,y), \quad y \text{ fixed} \\ v_x(y) &= f(x,y), \quad x \text{ fixed.} \end{aligned}$$

We then observe that $\frac{\mathrm{d}^m}{\mathrm{d}x^m} u_y(x) = \frac{\partial^m}{\partial x^m} f(x,y)$ and $\frac{\mathrm{d}^m}{\mathrm{d}y^m} v_x(y) = \frac{\partial^m}{\partial y^m} f(x,y)$ for all $m \geq 0$ provided the partial derivatives exist and are well defined. Consequently

$$\begin{aligned} u_y(x+h_1) &= u_y(x) + h_1 u_y'(x) + \frac{h_1^2}{2} u_y''(\xi) \\ &= f(x,y) + h_1 \frac{\partial f}{\partial x}(x,y) + \frac{h_1^2}{2} \frac{\partial^2 f}{\partial x^2}(\xi, y) \end{aligned}$$

and

$$\begin{aligned} v_x(y+h_2) &= v_x(y) + h_2 v_x'(y) + \frac{h_2^2}{2} v_x''(\eta) \\ &= f(x,y) + h_2 \frac{\partial f}{\partial y}(x,y) + \frac{h_2^2}{2} \frac{\partial^2 f}{\partial y^2}(x, \eta). \end{aligned}$$

Here, ξ is a point between x and $x+h_1$ and η is a point between y and $y+h_2$. In the remainder of this exposition we will assume that the *perturbations* $|h_1|$ and $|h_2|$ are moderately sized lest the results be overly inaccurate.

We now wish to express the function value $f(x+h_1, y+h_2)$ in terms of the value of f and its (partial) derivatives at (x,y). This will lead to the bi-variate Taylor expansion of f. Doing Taylor expansion in the y direction first whilst keeping x fixed we find that

$$\begin{aligned} f(x+h_1, y+h_2) &= v_{x+h_1}(y+h_2) \\ &= v_{x+h_1}(y) + h_2 \frac{\mathrm{d}}{\mathrm{d}y} v_{x+h_1}(y) + \frac{h_2^2}{2} \frac{\mathrm{d}^2}{\mathrm{d}y^2} v_{x+h_1}(\eta). \end{aligned}$$

Rewriting this in terms of the bi-variate function f we then get

$$f(x+h_1, y+h_2) = f(x+h_1, y) + h_2 \frac{\partial}{\partial y} f(x+h_1, y) + \frac{h_2^2}{2} \frac{\partial^2}{\partial y^2} f(x+h_1, \eta).$$

Differentiating now *each term* with respect to x we find

$$\begin{aligned} f(x+h_1, y+h_2) = f(x,y) &+ h_1 \frac{\partial}{\partial x} f(x,y) + \frac{h_1^2}{2} \frac{\partial^2}{\partial x^2} f(\xi, y) + \\ &h_2 \frac{\partial}{\partial y} \big(f(x,y) + h_1 \frac{\partial}{\partial x} f(x,y) + \frac{h_1^2}{2} \frac{\partial^2}{\partial x^2} f(\xi, y) \big) + \\ &\frac{h_2^2}{2} \frac{\partial^2}{\partial y^2} \big(f(x,\eta) + h_1 \frac{\partial}{\partial x} f(x,y) + \frac{h_1^2}{2} \frac{\partial^2}{\partial x^2} f(\xi, \eta) \big). \end{aligned}$$

Collecting terms in increasing powers of h_1 and h_2, we then see

$$f(x+h_1, y+h_2) = f(x,y) + h_1 \frac{\partial f}{\partial x}(x,y) + h_2 \frac{\partial f}{\partial y}(x,y) + \mathcal{O}(h^2). \tag{1}$$

Here, $h = \sqrt{h_1^2 + h_2^2}$ and $\mathcal{O}(h^2)$ signifies a function $g(x)$ such that

$$\lim_{h \to 0} \frac{g(h)}{h^2} = C, \quad 0 < |C| < \infty.$$

Note though, that the simplification (1) is a direct result of *disregarding* some of the information included in the "higher order terms".

Generalisation to several variables

The bi-variate Taylor expansion (1) may be further generalised to multi-variate functions, that is, functions of $n \geq 2$ variables, $f(x_1, x_2, \dots, x_n)$. Denoting by

$$\mathbf{x} = \left[x_1, x_2, \dots, x_n \right]^{\mathsf{T}}$$

a vector of n real components, we may make the notation for multi-variate functions more compact as

$$f(x_1, x_2, \ldots, x_n) \equiv f(\mathbf{x}).$$

We underscore though that this is merely a compact notation to signify that the function f depends on more than one scalar variable. Suppose now that $\mathbf{h} = \big[h_1, h_2, \ldots, h_n\big]^\mathsf{T}$ is a vector of (presumably small) 'displacements'. Then

$$f(\mathbf{x}+\mathbf{h}) = f(\mathbf{x}) + \sum_{j=1}^{n} h_j \frac{\partial f}{\partial x_j}(\mathbf{x}) + \mathcal{O}(h^2), \tag{2}$$

in which $h = \sqrt{\sum_{j=1}^n h_j^2}$, is the generalisation of (1) to functions of n scalar variables. We remark in particular that all of the partial derivatives $\partial f/\partial x_j$ are *evaluated* at the point $\mathbf{x}$. The above is certainly not a proof, but merely a statement of the general result.

Newton's method for systems of non-linear equations

We seek a method for the resolution of the system of n non-linear equations in n variables given by

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0. \end{aligned} \tag{3}$$

In other words, n non-linear equations each depending on n variables must be satisfied simultaneously. We will assume throughout the derivation that some point $\mathbf{r} = \big[r_1, r_2, \ldots, r_n\big]^\mathsf{T} \in \mathbb{R}^n$ exists for which all the equations are satisfied.

Our aim now is to define a *sequence* of *vectors* $\mathbf{x}_m \in \mathbb{R}^n$ with $m \geq 0$ and hopefully such that

$$\lim_{m\to\infty} \mathbf{x}_m = \mathbf{r}.$$

To this end, and to simplify the notation even further, define the *vector valued* function

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) \end{bmatrix}$$

The system of non-linear equations (3) may then be compactly written as

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

in which $\mathbf{0}$ denotes the zero vector, $\mathbf{0} = \big[0, 0, \ldots, 0\big]^\mathsf{T} \in \mathbb{R}^n$.

Now suppose $\mathbf{x}_m \approx \mathbf{r}$, and define $\mathbf{h} = \mathbf{r} - \mathbf{x}_m$. We wish to derive conditions on the vector $\mathbf{h}$ in order that $\mathbf{f}(\mathbf{r}) = \mathbf{0}$. Using the multi-variate Taylor expansion (2) in each of the n originial equations (3) we then get

$$\mathbf{0} = \mathbf{f}(\mathbf{r}) = \begin{bmatrix} f_1(\mathbf{x}_m+\mathbf{h}) \\ f_2(\mathbf{x}_m+\mathbf{h}) \\ \vdots \\ f_n(\mathbf{x}_m+\mathbf{h}) \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}_m) + \sum_{j=1}^n h_j \frac{\partial f_1}{\partial x_j}(\mathbf{x}_m) + \mathcal{O}(h^2) \\ f_2(\mathbf{x}_m) + \sum_{j=1}^n h_j \frac{\partial f_2}{\partial x_j}(\mathbf{x}_m) + \mathcal{O}(h^2) \\ \vdots \\ f_n(\mathbf{x}_m) + \sum_{j=1}^n h_j \frac{\partial f_n}{\partial x_j}(\mathbf{x}_m) + \mathcal{O}(h^2) \end{bmatrix}. \tag{4}$$

We now define the $n \times n$ *Jacobian matrix* $D\mathbf{f}(\mathbf{x})$ as

$$D\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \frac{\partial f_n}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{bmatrix},$$

or more succinctly, as $\big(D\mathbf{f}(\mathbf{x})\big)_{ij} = \frac{\partial f_i}{\partial x_j}(\mathbf{x})$ for all $i, j = 1, 2, \ldots, n$. Recalling briefly that the matrix-vector product $\mathbf{w} = A\mathbf{v}$ is defined component-wise as

$$w_i = \sum_{j=1}^n a_{ij} v_j$$

for all $i = 1, 2, \ldots, n$ (when $A \in \mathbb{R}^{n\times n}$ and $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$), we may rewrite (4) as

$$\mathbf{0} = \mathbf{f}(\mathbf{x}_m) + D\mathbf{f}(\mathbf{x}_m)\,\mathbf{h} + \mathbf{R}_m$$

in which the *residual* $\mathbf{R}_m \in \mathbb{R}^n$ is a vector whose components are all $\mathcal{O}(h^2)$.

Assuming now that the Jacobian matrix $D\mathbf{f}(\mathbf{x}_m)$ is regular (i.e. invertible), disregarding the residual $\mathbf{R}_m$ and *defining* the vector $\mathbf{h}_m$ as the (unique) solution to the $n \times n$ linear system

$$\mathbf{0} = \mathbf{f}(\mathbf{x}_m) + D\mathbf{f}(\mathbf{x}_m)\,\mathbf{h}_m \Leftrightarrow D\mathbf{f}(\mathbf{x}_m)\,\mathbf{h}_m = -\mathbf{f}(\mathbf{x}_m),$$

we arrive at the *next* term in the sequence $\{\mathbf{x}_m\}$ as

$$\mathbf{x}_{m+1} = \mathbf{x}_m + \mathbf{h}_m.$$

This, then, fully defines Newton's method for systems of non-linear equations as

1. Choose some initial point $\mathbf{x}_0$.
2. For all $m = 0, 1, 2, \ldots$ **until convergence**
   a) Compute the Jacobian matrix $J = D\mathbf{f}(\mathbf{x}_m)$.
   b) Solve the linear system $J\mathbf{h}_m = -\mathbf{f}(\mathbf{x}_m)$ with respect to $\mathbf{h}_m$.
   c) Set $\mathbf{x}_{m+1} = \mathbf{x}_m + \mathbf{h}_m$.

A few remarks on the method are in order. Most importantly we have to resolve an $n \times n$ linear system on **each** iteration of the method. Moreover, both the matrix and the right hand side in general *vary* as functions of $\mathbf{x}$, meaning we have to re-evaluate these quantities on **each** iteration too. This makes the Newton–Raphson method fairly expensive, especially for large systems (i.e. $n \gg 1$).

Finally, note the similarity to the original (scalar) Newton method

$$x_{m+1} = x_m - \frac{f(x_m)}{f'(x_m)}, \quad m = 0, 1, 2, \ldots \tag{5}$$

Defining h_m as the solution to the simple, linear equation

$$f'(x_m)\, h_m = -f(x_m)$$

for all $m = 0, 1, 2, \ldots$, equation (5) is exactly $x_{m+1} = x_m + h_m$ for all $m \geq 0$. We also notice that the Jacobian matrix $D\mathbf{f}(\mathbf{x}_m)$ plays the rôle of the derivative in the case of systems of non-linear equations.

Example

We consider the system of two equations given by

$$\begin{aligned} x_1 - x_2 + 1 &= 0 \\ x_1^2 + x_2^2 - 4 &= 0. \end{aligned} \tag{6}$$

Using vector notation this is $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ in which $\mathbf{x} = [x_1, x_2]^\mathsf{T}$ and the vector function $\mathbf{f}(\mathbf{x})$ is given by $\mathbf{f}(\mathbf{x}) = [x_1 - x_2 + 1, x_1^2 + x_2^2 - 4]^\mathsf{T}$. Graphically, solving

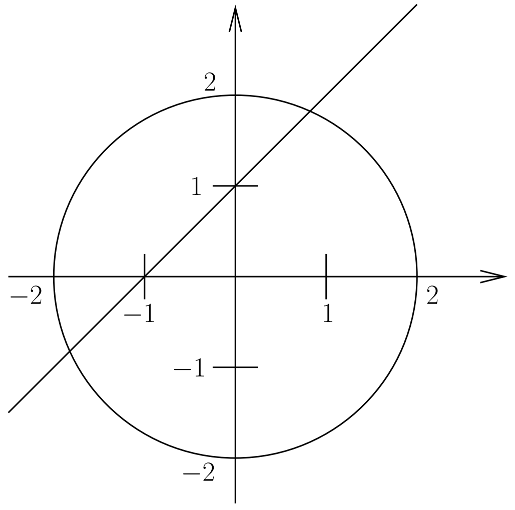

Figure 1: Graphical interpretation of the system (6).

this system corresponds to constructing the points at which the line $x_2 = x_1 + 1$ intersects the circle $x_1^2 + x_2^2 = 2^2$. The situation is shown in Figure 1. We notice from Figure 1 that the system has a solution near the point $(x_1, x_2) = (0.8, 1.8)$. We will demonstrate how Newton's method converges to this solution.

The Jacobian matrix of the system (6) can be computed exactly, the result being

$$D\mathbf{f}(\mathbf{x}) = \begin{bmatrix} 1 & -1 \\ 2x_1 & 2x_2 \end{bmatrix}.$$

Note in particular that some of the matrix entries in this case are constant while others depend on the values of x_1 and x_2. This is quite common in practice.

Starting from $\mathbf{x}_0 = [0.8, 1.8]^\mathsf{T}$, we first compute the function $\mathbf{f}(\mathbf{x}_0)$ and the Jacobian matrix $D\mathbf{f}(\mathbf{x}_0)$ as

$$\mathbf{f}(\mathbf{x}_0) = \begin{bmatrix} 0 \\ -0.12 \end{bmatrix}, \quad D\mathbf{f}(\mathbf{x}_0) = \begin{bmatrix} 1 & -1 \\ 1.6 & 3.6 \end{bmatrix}.$$

Thus, the initial *displacement* vector $\mathbf{h}_0$ is the solution to the linear system

$$\begin{bmatrix} 1 & -1 \\ 1.6 & 3.6 \end{bmatrix} \mathbf{h}_0 = -\mathbf{f}(\mathbf{x}_0) = \begin{bmatrix} 0 \\ 0.12 \end{bmatrix}. \tag{7}$$

It is ***very*** important to remember that the right hand side of the *linear* system (7) is $-\mathbf{f}(\mathbf{x}_0)$ and not $\mathbf{f}(\mathbf{x}_0)$. In other words, the negative sign is entirely essential to the success of the method. Solving the linear system (7) we find

$$\mathbf{h}_0 \approx \begin{bmatrix} 0.0230769 \\ 0.0230769 \end{bmatrix},$$

from which the next Newton iterate is

$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{h}_0 = \begin{bmatrix} 0.8 \\ 1.8 \end{bmatrix} + \begin{bmatrix} 0.0230769 \\ 0.0230769 \end{bmatrix} = \begin{bmatrix} 0.8230769 \\ 1.8230769 \end{bmatrix}.$$

To start the next Newton iteration, we compute the vector $\mathbf{f}(\mathbf{x}_1)$ and the Jacobian matrix $D\mathbf{f}(\mathbf{x}_1)$ as

$$D\mathbf{f}(\mathbf{x}_1) = \begin{bmatrix} 1.0000000 & -1.0000000 \\ 1.6461538 & 3.6461538 \end{bmatrix}, \quad \mathbf{f}(\mathbf{x}_1) = \begin{bmatrix} 0.0000000 \\ 0.0010651 \end{bmatrix}.$$

The second displacement vector, $\mathbf{h}_1$, is then the solution to the linear system

$$\begin{bmatrix} 1.0000000 & -1.0000000 \\ 1.6461538 & 3.6461538 \end{bmatrix} \mathbf{h}_1 = \underbrace{\begin{bmatrix} 0.0000000 \\ -0.0010651 \end{bmatrix}}_{=-\mathbf{f}(\mathbf{x}_1)}. \tag{8}$$

We then find that

$$\mathbf{h}_1 = \begin{bmatrix} -2.0125224 \cdot 10^{-4} \\ -2.0125224 \cdot 10^{-4} \end{bmatrix}, \quad \mathbf{x}_2 = \mathbf{x}_1 + \mathbf{h}_1 = \begin{bmatrix} 0.8228757 \\ 1.8228757 \end{bmatrix}.$$

Repeating this proces once more we find the final iterate

$$\mathbf{x}_3 = \begin{bmatrix} 0.82287565553230 \\ 1.82287565553230 \end{bmatrix}$$

which is correct to 16 decimal digits as you may check from the exact solution

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{r} = \begin{bmatrix} \frac{1}{2}(\sqrt{7}-1) \\ \frac{1}{2}(\sqrt{7}+1) \end{bmatrix}.$$

Implementing Newton's method in MATLAB

Solving all the linear systems needed in Newton's method, not to mention actually computing the matrices and right hand sides in the first place, quickly

becomes quite tedious—especially if the number of Newton iterations required for convergence is large. Fortunately, MATLAB expediently resolves linear systems numerically by means of the "matrix left division operator", `\`.

If we assume the existence of two problem-dependent functions `f` and `Df` which compute the values of $\mathbf{f}(\mathbf{x})$ and $D\mathbf{f}(\mathbf{x})$ respectively, the Newton method reduces to the two statements

```
h = - Df(x) \ f(x)
x = x + h
```

These statements can in fact be written in a single line as

```
h = - Df(x) \ f(x), x = x + h
```

The Newton process leading to the solution of the system (6) may thus be realised as

```
f  = @(x) [x(1) - x(2) + 1; x(1)^2 + x(2)^2 - 4];
Df = @(x) [1, -1; 2*x(1), 2*x(2)];

x  = [0.8; 1.8];    % initial value
h  = - Df(x) \ f(x), x = x + h
h  = - Df(x) \ f(x), x = x + h
h  = - Df(x) \ f(x), x = x + h
h  = - Df(x) \ f(x), x = x + h
```

and we observe that in the last step `h = [0; 0]` meaning convergence to the exact solution in four steps when starting from

$$\mathbf{x}_0 = \begin{bmatrix} 0.8 \\ 1.8 \end{bmatrix}.$$