PHY256 Notes on Newton-Raphson method, the Newton map, and solving non-linear systems of equations

A. C. Quillen

February 16, 2021

Contents

1 Newton's method

Given a differentiable function $f(x)$ on real numbers how can we find its roots? Newton's method is a iteration procedure for converging on a root. The idea is to make a map, N_f using f so that if we keep applying the map to an initial value of x we will converge on a root. We want the map to have orbits that converge on to roots.

The map can have interesting dynamics. For example, convergence is not assured. There may be no roots. In this case the orbits of the map can be chaotic or go to infinity. If there are more than one root, the basins of attraction for convergence to each root may be difficult to determine. The map does not always converge onto the nearest root. However, if there is a nearby root, then Newton's map converges very quickly, quadratically quickly in fact.

We start with a point $x_0, f(x_0)$ and compute the tangent of the function at this point. We look at the tangent line (with slope $f'(x_0)$) that goes through $x_0, f(x_0)$ and find where it crosses the y axis. The line has equation

$$y - y_0 = f'(x_0)(x - x_0) = y - f(x_0) = f'(x_0)(x - x_0)$$

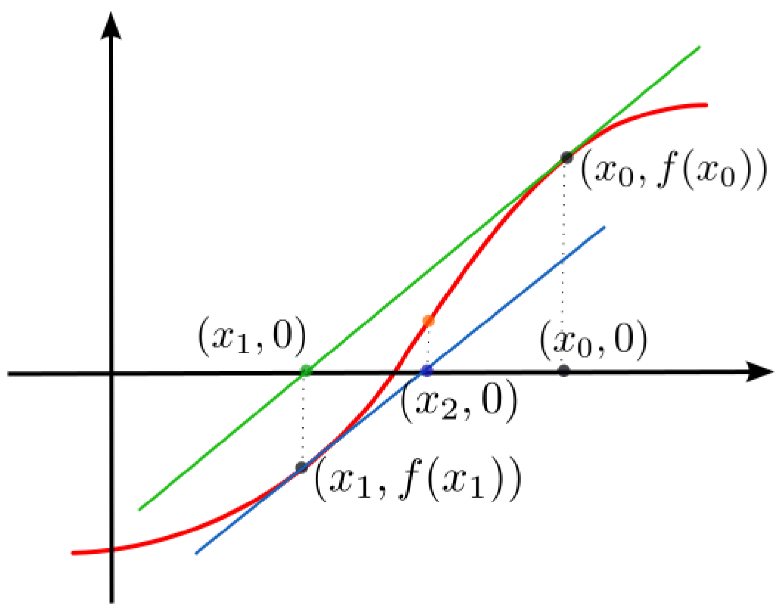

Figure 1: Illustration of Newton's method for converging to a root of a function. The function f is shown in red. The initial choice is x_0. Applying Newton's method then gives x_1 and then x_2. The iterated value depends on the slope of the function f.

and it crosses the y axis at $y = 0$ and where

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

We construct a map

$$x_{n+1} = N_f(x_n)$$

with

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

We can call N_f Newton's function for f.

Example Let $f(x) = x^3 - x$. The roots are $-1, 0, 1$. We compute

$$N_f(x) = x - \frac{x^3 - x}{3x^2 - 1} = \frac{2x^3}{3x^2 - 1}$$

The function is well behaved except at $x = 1/\sqrt{3}$. Near roots, the Newton map converges very quickly. But there are initial conditions that give problems (chaotic behavior) and those for which convergence is very slow (near $x = 1/\sqrt{3}$).

1.1 Quadratic functions

Let $f(x) = ax^2 + bx + c$ and $q(x) = x^2 - A$. The two functions N_f, and N_q are topologically conjugate using the function $h(x) = 2ax + b$ and with $A = b^2 - 4ac$. We can show that this is true by showing that $h \circ N_f = N_f \circ h$. We compute $N_f(x) = \frac{ax^2 - c}{2ax+b}$ and $N_q(x) = \frac{x^2+A}{2x}$.

$$h \circ N_f = 2a\frac{ax^2 - c}{2ax + b} + b = \frac{2a^2x^2 + 2abx + b^2 - 2ac}{2ax + b}$$

$$N_p \circ h = \frac{(2ax + b)^2 + A}{2(2ax + b)} = \frac{2a^2x^2 + 2abx + b^2 - 2ac}{2ax + b}$$

As a consequence, study of Newton's method on quadratic functions need only consider two cases $f(x) = x^2 - 1$ which has real roots and $f(x) = x^2 + 1$ which does not.

When $f(x) = x^2 + 1$ we find $N_f = (x^2 - 1)/2x$ which is pretty badly behaved near $x = 0$. In fact, orbits are chaotic. There is a topological conjugacy via $h(x) = \cot(x/2)$ to the map on the unit circle $D(\theta) = 2\theta$ which is chaotic.

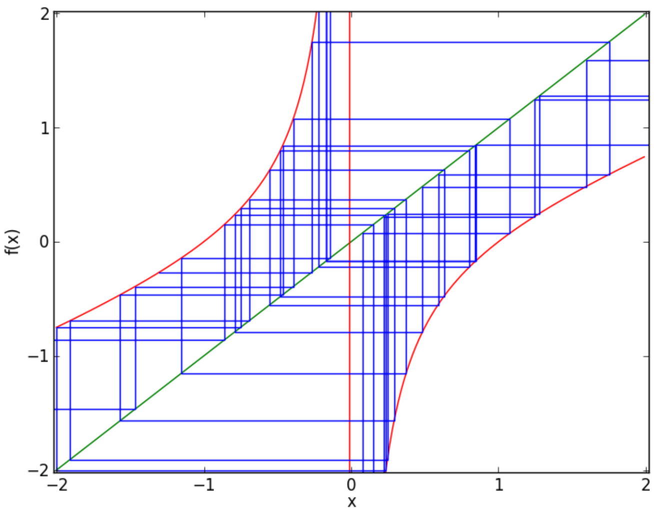

Figure 2: A cobweb plot showing trajectory for Newtons map $N_f(x) = x - (x^2 - 1)/(2x)$ for the function $f(x) = x^2 + 1$ which has no roots. The orbit is chaotic.

1.2 Convergence of Newton's map

Consider the difference

$$\delta_n \equiv |x_{n+1} - x_n|$$

that describes a distance away from convergence. If the map converges δ_n should decrease. As we get closer to a root this difference should decrease. We can write this difference

$$\delta_n = \left| x_n - \frac{f(x_n)}{f'(x_n)} - x_n \right| = \left| \frac{f(x_n)}{f'(x_n)} \right|$$

Let us define a function

$$r(x) = \frac{f(x)}{f'(x)}$$

so that

$$\delta_n = |r(x_n)|$$

Now we compute the next difference

$$\begin{aligned} \delta_{n+1} &= |r(x_n + \delta_n)| \\ &= |r(x_n + r(x_n))| \\ &= \left| r(x_n) + r'(x_n) r(x_n) \right| \\ &= \left| r(x_n)(1 + r'(x_n)) \right| \end{aligned}$$

$$r'(x) = \frac{f'(x)}{f'(x)} - \frac{f(x) f''(x)}{f'(x)^2} = 1 - \frac{f(x) f''(x)}{f'(x)^2}$$

$$\begin{aligned} 1 + r'(x) &= -\frac{f(x) f''(x)}{f'(x)^2} \\ &= -r(x) \frac{f''(x)}{f'(x)} \end{aligned}$$

Inserting this back into δ_{n+1}

$$\begin{aligned} \delta_{n+1} &= \left| r(x_n) r(x_n) \frac{f''(x_n)}{f'(x_n)} \right| \\ &= \delta_n^2 \left| \frac{f''(x_n)}{f'(x_n)} \right| \end{aligned}$$

This expressions shows that the distance to the root should decrease quadratically. This explains why when one is near a root, convergence is rapid and the technique is quite efficient.

1.3 Newton's method as a first order approximation

We want x^* such that $f(x*) = 0$. But we are starting at different value x_0. We can search for δ such that

$$f(x_0 + \delta) = 0$$

Let us expand this to first order in δ.

$$f(x_0) + f'(x_0)\delta = 0$$

To first order

$$\delta = -\frac{f(x_0)}{f'(x_0)}$$

or

$$x_1 = x_0 + \delta = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Here x_1 is the solution to the first order approximation.

Remark Newton's method appears to be a simple root finding technique. However it can be applied to functions and estimates of its convergence rate are useful.

2 Solving systems of equations

Consder a linear set of equations that can be put in matrix form

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

where $\mathbf{A}$ is a matrix of dimension $n \times n$, and $\mathbf{x}, \mathbf{b}$ are vectors of length n. If the matrix is invertible the set of equations can be solved with

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

This is equivalent to finding the root of the equation

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b} = 0$$

Below we will give an example of how to solve a non-linear system of equations iteratively using Newton's method and by solving a set of linear equations. Simultaneously we illustrate the use of linear algebra for multi-dimensional root finding. The multidimensional version of Newton's map gives a numerical method for iterative solution of a system of non-linear system equations.

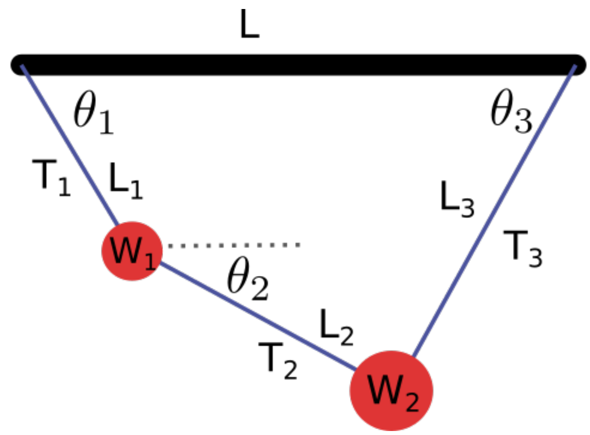

Figure 3: Two masses of weights W_1, W_2 suspended from a bar of length L by three lengths of string L_1, L_2, L_3. The tensions on each piece of string are T_1, T_2, T_3. The angles between horizontal and string orientations are $\theta_1, \theta_2, \theta_3$.

2.1 Two Masses on a string

We have two weights W_1, W_2, connected by three pieces of string. The ends of the string are suspected from a bar that has length L. The three pieces of string have lengths L_1, L_2, L_3. The angle between A steady state has angles $\theta_1, \theta_2, \theta_3$ as shown in Figure 3 and tensions T_1, T_2, T_3 on the three pieces of string.

The known values are $W_1, W_2, L, L_1, L_2, L_3$. The variables that we want to find are $T_1, T_2, T_3, \theta_1, \theta_2, \theta_3$.

Constraints on horizontal lengths

$$\begin{aligned} L_1 \cos\theta_1 + L_2 \cos\theta_2 + L_3 \cos\theta_3 &= L \\ L_1 \sin\theta_1 + L_2 \sin\theta_2 - L_3 \sin\theta &= 0 \end{aligned}$$

Trigonometric identities

$$\begin{aligned} \cos^2\theta_1 + \sin^2\theta_1 &= 0 \\ \cos^2\theta_2 + \sin^2\theta_2 &= 0 \\ \cos^2\theta_3 + \sin^2\theta_3 &= 0 \end{aligned}$$

Constraints on Forces

$$
\begin{aligned}
T_1 \sin\theta_1 - T_2 \sin\theta_2 &= W_1 \\
T_1 \cos\theta_1 - T_2 \cos\theta_2 &= 0 \\
T_2 \sin\theta_2 + T_3 \sin\theta_3 &= W_2 \\
T_2 \sin\cos_2 - T_3 \cos\theta_3 &= 0
\end{aligned}
$$

We chose a state-vector

$$
\mathbf{y} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix} = \begin{pmatrix} \sin\theta_1 \\ \sin\theta_2 \\ \sin\theta_3 \\ \cos\theta_1 \\ \cos\theta_2 \\ \cos\theta_3 \\ T_1 \\ T_2 \\ T_3 \end{pmatrix} \tag{1}
$$

The system of equations we can write as

$$
\mathbf{f}(\mathbf{y}) = 0 \tag{2}
$$

with

$$
\mathbf{f}(\mathbf{y}) = \begin{pmatrix} f_1(\mathbf{y}) \\ f_2(\mathbf{y}) \\ f_3(\mathbf{y}) \\ f_4(\mathbf{y}) \\ f_5(\mathbf{y}) \\ f_6(\mathbf{y}) \\ f_7(\mathbf{y}) \\ f_8(\mathbf{y}) \\ f_9(\mathbf{y}) \end{pmatrix} = \begin{pmatrix} L_1 x_4 + L_2 x_5 + L_3 x_6 - L \\ L_1 x_1 + L_2 x_2 - L_3 x_3 \\ x_7 x_1 - x_8 x_2 - W_1 \\ x_7 x_4 - x_8 x_5 \\ x_8 x_2 + x_9 x_3 - W_2 \\ x_8 x_5 - x_9 x_6 \\ x_1^2 + x_4^2 - 1 \\ x_2^2 + x_5^2 - 1 \\ x_3^2 + x_6^2 - 1 \end{pmatrix} = 0 \tag{3}
$$

We notice that the constraints are non-linear. This means we cannot simply solve the set by inverting a matrix or using a matrix solve. However we can start with a guess $\mathbf{y}_0$ and iteratively find a solution.

Starting with our initial guess, we would have $\mathbf{f}(\mathbf{y}_0) \neq 0$ let $\mathbf{y}_1 = \mathbf{y}_0 + \Delta\mathbf{y}$. We expand

$$
\mathbf{f}(\mathbf{y}_1) = \mathbf{f}(\mathbf{y}_0) + \mathbf{F}'\Delta\mathbf{y}
$$

where $\mathbf{F}'$ is the Jacobian evaluated at $\mathbf{y}_0$. We set the result to zero giving

$$
\mathbf{f}(\mathbf{y}_1) = \mathbf{f}(\mathbf{y}_0) + \mathbf{F}'\Delta\mathbf{y} = 0
$$

This can be recognized as a linear matrix equation which we can solve for $\Delta\mathbf{y}$.

Lets be more specific about what we mean by this equation

$$\begin{pmatrix} f_1(\mathbf{y}_0) \\ f_2(\mathbf{y}_0) \\ \vdots \\ f_9(\mathbf{y}_0) \end{pmatrix} + \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_9} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_9} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_9}{\partial x_1} & \frac{\partial f_9}{\partial x_2} & \cdots & \frac{\partial f_9}{\partial x_9} \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_9 \end{pmatrix} = 0. \tag{4}$$

The partial derivatives are also evaluated at $\mathbf{y}_0$. We can solve this linear set of equations

$$\Delta\mathbf{y} = -\mathbf{F}'^{-1}\mathbf{f}(\mathbf{y}_0)$$

where the matrix $\mathbf{F}'^{-1}$ is the inverse of the Jacobian matrix evaluated at $\mathbf{y}_0$. We update our state vector with

$$\mathbf{y}_1 = \mathbf{y}_0 + \Delta\mathbf{y} = \mathbf{y}_0 - \mathbf{F}'^{-1}|_{\mathbf{y}_0}\mathbf{f}(\mathbf{y}_0)$$

The new state vector y_1 should be closer to a root or steady state solution for the mass/string system. The form of the equation resembles the Newton map!

The procedure is to iteratively find new and better values of $\mathbf{y}$ by applying the Newton method (sometimes called the Newton-Raphson method) in matrix form.

For this to work, a good initial guess $\mathbf{y_0}$ is required. There could be local minima and if the initial guess is not near enough to the one you want, the technique might not converge or might converge on an undesirable local minimum. To make the method robust, a procedure for choosing a step must be in place if the Jacobian matrix cannot be inverted.