Newton's method and high order iterations

Pascal Sebah and Xavier Gourdon
numbers.computation.free.fr/Constants/constants.html

October 3, 2001

Abstract

An introduction to the famous method introduced by Newton to solve non linear equations. It includes a description of higher order methods (cubic and more).

1 Introduction

It is of great importance to solve equations of the form

$$f(x) = 0,$$

in many applications in Mathematics, Physics, Chemistry, ... and of course in the computation of some important mathematical constants or functions like square roots. In this essay, we are only interested in one type of methods : the *Newton's methods.*

1.1 Newton's approach

Around 1669, Isaac Newton (1643-1727) gave a new algorithm [3] to solve a polynomial equation and it was illustrated on the example $y^3 - 2y - 5 = 0$. To find an accurate root of this equation, first one must guess a starting value, here $y \approx 2$. Then just write $y = 2 + p$ and after the substitution the equation becomes

$$p^3 + 6p^2 + 10p - 1 = 0.$$

Because p is supposed to be small we neglect $p^3 + 6p^2$ compared to $10p - 1$ and the previous equation gives $p \approx 0.1$, therefore a better approximation of the root is $y \approx 2.1$. It's possible to repeat this process and write $p = 0.1 + q$, the substitution gives

$$q^3 + 6.3q^2 + 11.23q + 0.061 = 0,$$

hence $q \approx -0.061/11.23 = -0.0054...$ and a new approximation for $y \approx 2.0946...$ And the process should be repeated until the required number of digits is reached.

In his method, Newton doesn't explicitly use the notion of derivative and he only applies it on polynomial equations.

1.2 Raphson's iteration

A few years later, in 1690, a new step was made by Joseph Raphson (1678-1715) who proposed a method [6] which avoided the substitutions in Newton's approach. He illustrated his algorithm on the equation $x^3 - bx + c = 0$, and starting with an approximation of this equation $x \approx g$, a better approximation is given by

$$x \approx g + \frac{c + g^3 - bg}{b - 3g^2}.$$

Observe that the denominator of the fraction is the opposite of the derivative of the numerator.

This was the historical beginning of the very important Newton's (or sometimes called Newton-Raphson's) algorithm.

1.3 Later studies

The method was then studied and generalized by other mathematicians like Simpson (1710-1761), Mourraille (1720-1808), Cauchy (1789-1857), Kantorovich (1912-1986), ... The important question of the choice of the starting point was first approached by Mourraille in 1768 and the difficulty to make this choice is the main drawback of the algorithm (see also [5]).

2 Newton's method

Nowadays, Newton's method is a generalized process to find an accurate root of a system (or a single) equations $f(x) = 0$. We suppose that f is a C^2function on a given interval, then using Taylor's expansion near x

$$f(x + h) = f(x) + hf'(x) + O(h^2),$$

and if we stop at the first order (linearization of the equation), we are looking for a small h such as

$$f(x + h) = 0 \approx f(x) + hf'(x),$$

giving

$$h = -\frac{f(x)}{f'(x)},$$
$$x + h = x - \frac{f(x)}{f'(x)}.$$

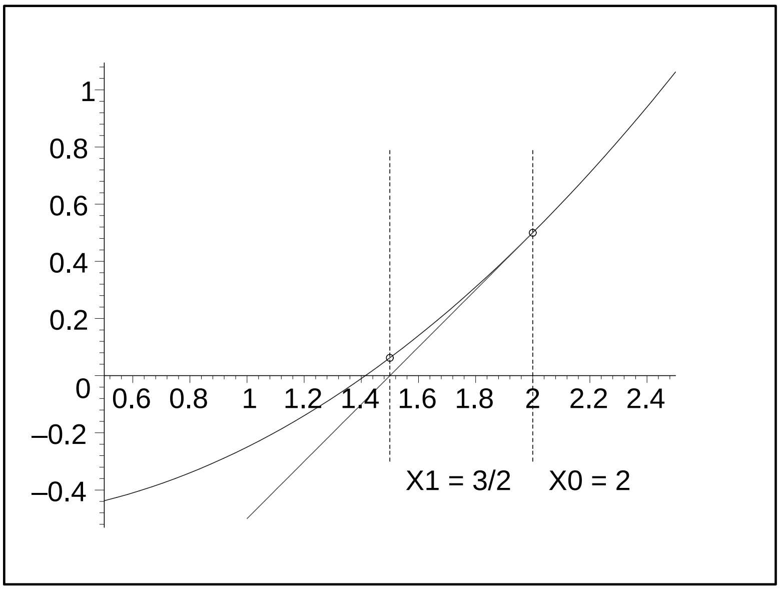

Figure 1: One Iteration of Newton's method

2.0.1 Newton's iteration

The Newton iteration is then given by the following procedure: start with an initial guess of the root x_0, then find the limit of recurrence:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

and Figure 1 is a geometrical interpretation of a single iteration of this formula.

Unfortunately, this iteration may not converge or is unstable (we say *chaotic* sometimes) in many cases. However we have two important theorems giving conditions to insure the convergence of the process (see [2]).

2.0.2 Convergence's conditions

Theorem 1 *Let x^* be a root of $f(x) = 0$, where f is a C^2function on an interval containing x^*, and we suppose that $|f'(x^*)| > 0$, then Newton's iteration will converge to x^* if the starting point x_0 is close enough to x^*.*

This theorem insure that Newton's method will always converge if the initial point is sufficiently close to the root and if this root if not singular (that is $f'(x^*)$ is non zero). This process has the *local convergence* property. A more constructive theorem was given by Kantorovich, we give it in the particular case of a function of a single variable x.

Theorem 2 *(Kantorovich). Let f be a C^2 numerical function from an open interval I of the real line, let x_0 be a point in this interval, then if there exist finite positive constants (m_0, M_0, K_0) such as*

$$\left|\frac{1}{f'(x_0)}\right| < m_0,$$
$$\left|\frac{f(x_0)}{f'(x_0)}\right| < M_0,$$
$$|f''(x_0)| < K_0,$$

and if $h_0 = 2m_0M_0K_0 \leq 1$,Newton's iteration will converge to a root x^* of $f(x) = 0$, and

$$|x_n - x^*| < 2^{1-n} M_0 h_0^{2^{n-1}}.$$

This theorem gives sufficient conditions to insure the existence of a root and the convergence of Newton's process. More if $h_0 < 1$, the last inequality shows that the convergence is quadratic (the number of good digits is doubled at each iteration). Note that if the starting point x_0 tends to the root x^*, the constant M_0 will tend to zero and h_0 will become smaller than 1, so the local convergence theorem is a consequence of Kantorovich's theorem.

3 Cubic iteration

Newton's iteration may be seen as a first order method (or linearization method), it's possible to go one step further and write the Taylor expansion of f to a higher order

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + O(h^3),$$

and we are looking for h such as

$$f(x+h) = 0 \approx f(x) + hf'(x) + \frac{h^2}{2}f''(x),$$

we take the smallest solution for h (we have to suppose that $f'(x)$ and $f''(x)$ are non zero)

$$h = -\frac{f'(x)}{f''(x)}\left(1 - \sqrt{1 - \frac{2f(x)f''(x)}{f'(x)^2}}\right).$$

It's not necessary to compute the square root, because if $f(x)$ is small, using the expansion

$$1-\sqrt{1-\alpha}=\frac{\alpha}{2}+\frac{\alpha^2}{8}+O(\alpha^3),$$

h becomes

$$h=-\frac{f(x)}{f'(x)}\left(1+\frac{f(x)f''(x)}{2f'(x)^2}+...\right).$$

The first attempt to use the second order expansion is due to the astronomer E. Halley (1656-1743) in 1694.

3.0.3 Householder's iteration

The previous expression for h, allows to derive the following cubic iteration (the number of digits triples at each iteration), starting with x_0

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}\left(1+\frac{f(x_n)f''(x_n)}{2f'(x_n)^2}\right).$$

This procedure is given in [1]. It can be efficiently used to compute the inverse or the square root of a number.

Another similar cubic iteration may be given by

$$x_{n+1}=x_n-\frac{2f(x_n)f'(x_n)}{2f'(x_n)^2-f(x_n)f''(x_n)},$$

sometimes known as Halley's method. We may also write it as

$$\begin{aligned}x_{n+1}&=x_n-f(x_n)\left(\frac{1}{f'(x_n)-f(x_n)f''(x_n)/(2f'(x_n))}\right)\\&=x_n-\frac{f(x_n)}{f'(x_n)}\left(1-\frac{f(x_n)f''(x_n)}{2f'(x_n)^2}\right)^{-1}.\end{aligned}$$

Note that if we replace $(1-\alpha)^{-1}$ by $1+\alpha+O(\alpha^2)$, we retrieve Householder's iteration.

4 High order iterations

4.1 Householder's methods

Under some conditions of regularity of f and it's derivative, Householder gave in [1] the general iteration

$$x_{n+1}=x_n+(p+1)\left(\frac{(1/f)^{(p)}}{(1/f)^{(p+1)}}\right)_{x_n},$$

where p is an integer and $(1/f)^{(p)}$ is the derivative of order p of the inverse of the function f. This iteration has convergence of order $(p+2)$. For example

$p = 0$ has quadratic convergence (order 2) and the formula gives back Newton's iteration while $p = 1$ has cubical convergence (order 3) and gives again Halley's method. Just like Newton's method a good starting point is required to insure convergence.

Using the iteration with $p = 2$, gives the following iteration which has quartical convergence (order 4)

$$x_{n+1} = x_n - f(x_n)\left(\frac{f'(x_n)^2 - f(x_n)f''(x_n)/2}{f'(x_n)^3 - f(x_n)f'(x_n)f''(x_n) + f^{(3)}(x_n)f^2(x_n)/6}\right).$$

4.2 Modified methods

Another idea is to write

$$x_{n+1} = x_n + h_n + a_2^n\frac{h_n^2}{2!} + a_3^n\frac{h_n^3}{3!} + ...,$$

where $h_n = -f(x_n)/f'(x_n)$ is given by the simple Newton's iteration and $(a_2^n, a_3^n, ...)$ are real parameters which we will estimate in order to minimize the value of $f(x_{n+1})$:

$$f(x_{n+1}) = f\left(x_n + h_n + a_2^n\frac{h_n^2}{2!} + a_3^n\frac{h_n^3}{3!} + ...\right),$$

we assume that f is regular enough and $h_n + a_2^n h_n^2/2! + a_3^n h_n^3/3! + ...$ is small, hence using the expansion of f near x_n

$$f(x_{n+1}) = f(x_n) + \left(h_n + a_2^n\frac{h_n^2}{2!} + a_3^n\frac{h_n^3}{3!} + ...\right)f'(x_n) + \left(h_n + a_2^n\frac{h_n^2}{2!} + a_3^n\frac{h_n^3}{3!} + ...\right)^2\frac{f''(x_n)}{2} + ...,$$

and because $f(x_n) + h_n f'(x_n) = 0$, we have

$$f(x_{n+1}) = (a_2^n f'(x_n) + f''(x_n))\frac{h_n^2}{2!} + \left(a_3^n f'(x_n) + 3a_2^n f''(x_n) + f^{(3)}(x_n)\right)\frac{h_n^3}{3!} + O(h_n^4).$$

A good choice for the a_i^n is clearly to cancel as many terms as possible in the previous expansion, so we impose

$$
\begin{aligned}
a_2^n &= -\frac{f_n''}{f_n'},\\
a_3^n &= \frac{-f_n' f_n^{(3)} + 3\left(f_n''\right)^2}{\left(f_n'\right)^2},\\
a_4^n &= \frac{-\left(f_n'\right)^2 f_n^{(4)} + 10 f_n' f_n'' f_n^{(3)} - 15\left(f_n''\right)^3}{\left(f_n'\right)^3},\\
a_5^n &= \frac{-\left(f_n'\right)^3 f_n^{(5)} + 15\left(f_n'\right)^2 f_n'' f_n^{(4)} + 10\left(f_n'\right)^2\left(f_n^{(3)}\right)^2 - 105 f_n'\left(f_n''\right)^2 f_n^{(3)} + 105\left(f_n''\right)^4}{\left(f_n'\right)^4},\\
a_6^n &= ...
\end{aligned}
$$

with, for the sake of brevity, the notation $f_n^{(k)} = f^{(k)}(x_n)$. The formal values of the a_i^n may be computed for much larger values of i. Finally the general iteration is

$$
\begin{aligned}
x_{n+1} &= x_n + h_n\left(1 + a_2^n \frac{h_n}{2!} + a_3^n \frac{h_n^2}{3!} + a_4^n \frac{h_n^3}{4!} + ...\right)\\
&= x_n - \frac{f(x_n)}{f'(x_n)}\left(1 + \frac{f''(x_n)}{2! f'(x_n)}\left(\frac{f(x_n)}{f'(x_n)}\right) + \frac{3f''(x_n)^2 - f'(x_n) f^{(3)}(x_n)}{3! f'(x_n)^2}\left(\frac{f(x_n)}{f'(x_n)}\right)^2 + ...\right).
\end{aligned}
$$

For example, if we stop at a_3^n and set $a_4^n = a_5^n = ... = 0$, we have the helpful *quartic modified iteration* (note that this iteration is different than the previous Householder's quartic method)

$$
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\left(1 + \frac{f(x_n) f''(x_n)}{2! f'(x_n)^2} + \frac{f(x_n)^2\left(3f''(x_n)^2 - f'(x_n) f^{(3)}(x_n)\right)}{3! f'(x_n)^4}\right),
$$

and if we omit a_3^n, we retrieve Householder's cubic iteration.

It's also possible to find the expressions for $(a_4^n, a_5^n, a_6^n, a_7^n, ...)$, and define *quintic, sextic, septic, octic* ... iterations.

5 Examples

In the essay on the square root of two, some applications of those methods are given and also in the essay on how compute the inverse or the square root of a number. Here let's compare the different iterations on the computation of the *Golden Ratio*

$$
\varphi = \frac{1 + \sqrt{5}}{2}.
$$

First we write $\varphi = 1/2 + x$ and $x = \sqrt{5}/2$ is root of the equation

$$\frac{1}{x^2} - \frac{4}{5} = 0,$$

then, it's convenient to set $h_n = 1 - 4/5 \times x_n^2$, we have the 5 algorithms, deduced from the general previous methods (left as exercise!), with respectively quadratic, cubic, quartic, sextic and octic convergence :

$$\begin{aligned}
x_{n+1} &= x_n + \frac{1}{2}x_n h_n \qquad \text{(Newton)},\\
x_{n+1} &= x_n + \frac{1}{8}x_n h_n\,(4 + 3h_n) \qquad \text{(Householder)},\\
x_{n+1} &= x_n + \frac{1}{16}x_n h_n\left(8 + 6h_n + 5h_n^2\right) \qquad \text{(Quartic)},\\
x_{n+1} &= x_n + \frac{1}{256}x_n h_n\left(128 + 96h_n + h_n^2\left(80 + 70h_n + 63h_n^2\right)\right) \qquad \text{(Sextic)},\\
x_{n+1} &= x_n + \frac{1}{2048}x_n h_n\left(1024 + 768h_n + h_n^2\left(640 + 560h_n + h_n^2\left(504 + 462h_n + 429h_n^2\right)\right)\right) \qquad \text{(Octic)}.
\end{aligned}$$

Starting with the initial value $x_0 = 1.118$, the first iteration becomes respectively

$$\begin{aligned}
\varphi_1^{Newton} &= x_1 + 1/2 = 1.61803398(720...) \qquad \text{8 digits},\\
\varphi_1^{Householder} &= x_1 + 1/2 = 1.6180339887498(163...) \qquad \text{13 digits},\\
\varphi_1^{Quartic} &= x_1 + 1/2 = 1.61803398874989484(402...) \qquad \text{17 digits},\\
\varphi_1^{Sextic} &= x_1 + 1/2 = 1.6180339887498948482045868(216...) \qquad \text{25 digits},\\
\varphi_1^{Octic} &= x_1 + 1/2 = 1.618033988749894848204586834365638(076...) \qquad \text{33 digits},
\end{aligned}$$

and one more step gives for x_2 respectively 17, 39, 69,154 and 273 good digits. If we iterate just three more steps, x_5 will give respectively 139, 1049, 4406, 33321 and 140053 correct digits ... Observe that h_n tends to zero as n increases which may be used to accelerate the computations and the choice of the best algorithm depends on your implementation (algorithm used for multiplication, the square, ...)

6 Newton's method for several variables

Newton's method may also be used to find a root of a system of two or more non linear equations

$$\begin{aligned}
f(x, y) &= 0\\
g(x, y) &= 0,
\end{aligned}$$

where f and g are C^2functions on a given domain. Using Taylor's expansion of the two functions near (x, y) we find

$$
\begin{aligned}
f(x+h, y+k) &= f(x, y) + h\frac{\partial f}{\partial x} + k\frac{\partial f}{\partial y} + O(h^2 + k^2) \\
g(x+h, y+k) &= g(x, y) + h\frac{\partial g}{\partial x} + k\frac{\partial g}{\partial y} + O(h^2 + k^2)
\end{aligned}
$$

and if we keep only the first order terms, we are looking for a couple (h, k) such as

$$
\begin{aligned}
f(x+h, y+k) &= 0 \approx f(x, y) + h\frac{\partial f}{\partial x} + k\frac{\partial f}{\partial y} \\
g(x+h, y+k) &= 0 \approx g(x, y) + h\frac{\partial g}{\partial x} + k\frac{\partial g}{\partial y}
\end{aligned}
$$

hence it's equivalent to the linear system

$$
\begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix} \begin{pmatrix} h \\ k \end{pmatrix} = -\begin{pmatrix} f(x, y) \\ g(x, y) \end{pmatrix}.
$$

The (2×2) matrix is called the *Jacobian matrix* (or *Jacobian*) and is sometimes denoted

$$
J(x, y) = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix}
$$

(it's generalization as a $(N \times N)$ matrix for a system of N equations and N variables is immediate) . This suggest to define the new process

$$
\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - J^{-1}(x_n, y_n) \begin{pmatrix} f(x_n, y_n) \\ g(x_n, y_n) \end{pmatrix}
$$

starting with an initial guess (x_0, y_0) and under certain conditions (which are not so easy to check and this is again the main disadvantage of the method), it's possible to show that this process converges to a root of the system. The convergence remains quadratic.

Example 3 *We are looking for a root near* $(x_0 = -0.6, y_0 = 0.6)$ *of the following system*

$$
\begin{aligned}
f(x, y) &= x^3 - 3xy^2 - 1 \\
g(x, y) &= 3x^2y - y^3,
\end{aligned}
$$

here the Jacobian and its inverse become

$$J(x_n, y_n) = 3 \begin{pmatrix} x_n^2 - y_n^2 & -2x_n y_n \\ 2x_n y_n & x_n^2 - y_n^2 \end{pmatrix}$$

$$J^{-1}(x_n, y_n) = \frac{1}{3\left(x_n^2 + y_n^2\right)^2} \begin{pmatrix} x_n^2 - y_n^2 & 2x_n y_n \\ -2x_n y_n & x_n^2 - y_n^2 \end{pmatrix}$$

and the process gives

$$\begin{aligned}
x_1 &= -0.40000000000000000000, y_1 = 0.86296296296296296296 \\
x_2 &= -0.50478978186242263605, y_2 = 0.85646430512069295697 \\
x_3 &= -0.49988539803643124722, y_3 = 0.86603764032215486664 \\
x_4 &= -0.50000000406150565266, y_4 = 0.86602539113638168322 \\
x_5 &= -0.49999999999999983928, y_5 = 0.86602540378443871965 \\
&...
\end{aligned}$$

Depending on your initial guess Newton's process may converge to one of the three roots of the system:

$$(-1/2, \sqrt{3}/2), (-1/2, -\sqrt{3}/2), (1, 0),$$

and for some values of (x_0, y_0) *the convergence of the process may be tricky! The study of the influence of this initial guess leads to aesthetic fractal pictures.*

Cubic convergence also exists for several variables system of equations : Chebyshev's method.

References

[1] A. S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, New York, (1970)

[2] L.V. Kantorovich and G.P. Akilov, *Functional Analysis in Normed Spaces*, Pergamon Press, Elmsford, New York, (1964)

[3] I. Newton, *Methodus fluxionum et serierum infinitarum*, (1664-1671)

[4] J.M. Ortega and W.C. Rheinboldt, *Iterative solution of non linear equations in several variables*, (1970)

[5] H. Qiu, *A Robust Examination of the Newton-Raphson Method with Strong Global Convergence Properties*, Master's Thesis, University of Central Florida, (1993)

[6] J. Raphson, *Analysis Aequationum universalis*, London, (1690)