Solution of Nonlinear Equations

(Com S 477/577 Notes)

Yan-Bin Jia

Sep 15, 2020

One of the most frequently occurring problems in scientific work is to find the roots of an equation of the form

$$f(x) = 0. \tag{1}$$

The function $f(x)$ may be given explicitly as, for example, a polynomial or a transcendental function. Frequently, however, $f(x)$ may be known only implicitly in that only a rule for evaluating it on any argument is known. In rare cases it may be possible to obtain the exact roots such as in the case of a factorizable polynomial. In general, however, we can hope to obtain only approximate values of the roots, relying on some computational techniques to produce the approximation. In this lecture, we will introduce some elementary iterative methods for finding a root of equation (1), in other words, a *zero* of $f(x)$.

1 Bisection

Suppose the function $f(x)$ is over the interval $[a_0, b_0]$ such that $f(a_0)f(b_0) \leq 0$. If f is well-behaved, then it will have a root between a_0 and b_0. We halve the interval $[a_0, b_0]$ while still bracketing the root, and repeat.

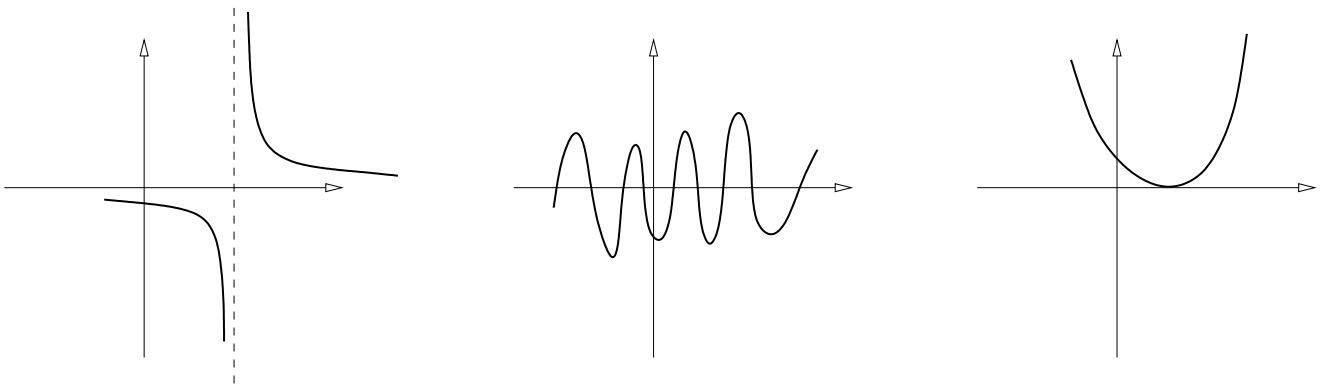
for $i = 0, 1, 2, \ldots,$ until satisfied, **do**
  $m \leftarrow (a_i + b_i)/2$
  if $f(a_i)f(m) \leq 0$
    then $a_{i+1} \leftarrow a_i$
      $b_{i+1} \leftarrow m$
    else $a_{i+1} \leftarrow m$
      $b_{i+1} \leftarrow b_i$

The first part of the idea is critical to many root-finding techniques, namely, to find an interval that *brackets* a root of f. This can be difficult though in a number of situations shown in the next figure.

(a) The interval straddles a singularity. In this case, bisection will converge to that singularity.

(b) Multiple roots are bracketed. Bisection will find only one while leaving an impression that no other roots lie in the interval.

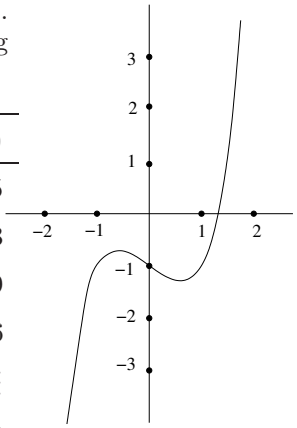
(c) A *double root* r, that is, $f(r) = f'(r) = 0$, is bracketed. Since $f(a_0)f(b_0) > 0$, bisection will not even be invoked.



(a) (b) (c)

EXAMPLE 1. $f(x) = x^3 - x - 1$

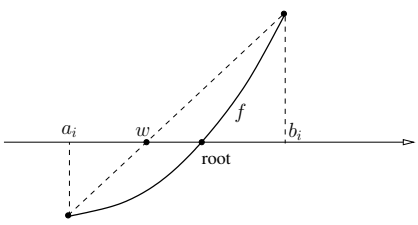
Since f is a cubic it has either one or three real zeros.[1] There is only one variation in the signs of its coefficients. Thus f must have only a single zero instead of three. This zero is initially bracketed by $[1, 2]$. The iteration results are given in the following table:



i	$[a_i, b_i]$	$f(a_i)$	$f(b_i)$	$\frac{a_i+b_i}{2}$	$f(\frac{a_i+b_i}{2})$
0	[1, 2]	-1	5	1.5	0.875
1	[1, 1.5]	-1	0.875	1.25	-0.29683
2	[1.25,1.5]	-0.296875	0.875	1.375	0.224609
3	[1.25, 1.375]	-0.296875	0.22460937	1.3125	-0.515136
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
14	[1.324707, 1.3247681]	$-4.659 \cdot 10^{-5}$	$2.137 \cdot 10^{-4}$	1.3247375	$8.355 \cdot 10^{-5}$
15	[1.324707, 1.3247375]	$-4.659 \cdot 10^{-5}$	$8.355 \cdot 10^{-5}$	1.3247223	$1.848 \cdot 10^{-5}$

In each step of the bisection method, the length of the bracketing interval is halved. Hence each step produces one more binary digit, or bit, in the approximation to the root. Bisection can be slow, but it is simple and robust. It is therefore sometimes used as a backup for more complicated algorithms.

2 Regula Falsi



The method of regular falsi uses the idea that it often makes sense to assume that the function is linear locally. Instead of using the midpoint of the bracketing interval to select a new root estimate, use a weighted average:

$$w = \frac{f(b_i)a_i - f(a_i)b_i}{f(b_i) - f(a_i)}. \tag{2}$$

[1] Root counting for polynomials will be introduced in an upcoming lecture.

Here $f(b_i)$ and $f(a_i)$ have opposite signs under bracketing. Note that w is just the weighted average of a_i and b_i with weights $|f(b_i)|$ and $|f(a_i)|$, that is

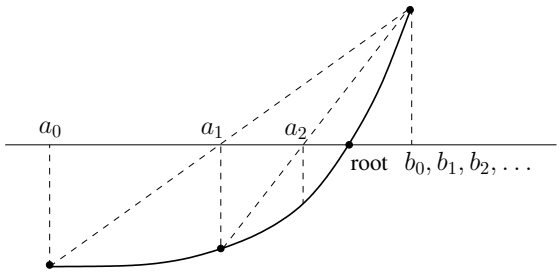
$$w = \frac{|f(b_i)|}{|f(b_i)| + |f(a_i)|} a_i + \frac{|f(a_i)|}{|f(b_i)| + |f(a_i)|} b_i. \tag{3}$$

If $|f(b_i)|$ is larger than $|f(a_i)|$, then the new root estimate w is closer to a_i than to b_i. as shown in the figure on the left.

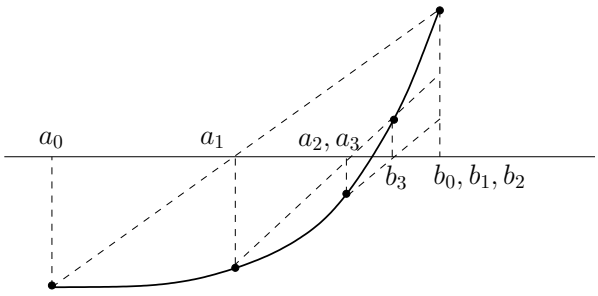
Indeed, the weighted average w is the intersection of the x-axis with the line through the points $(a_i, f(a_i))$ and $(b_i, f(b_i))$. Such a straight line is a *secant* to $f(x)$. The description of the regular falsi algorithm is similar to that of bisection:

for $i = 0, 1, 2, \ldots$, until satisfied, **do**
  $w \leftarrow (f(b_i)a_i - f(a_i)b_i)/(f(b_i) - f(a_i))$
  if $f(a_i)f(w) \leq 0$
    then $a_{i+1} \leftarrow a_i$
      $b_{i+1} \leftarrow w$
    else $a_{i+1} \leftarrow w$
      $b_{i+1} \leftarrow b_i$

A drawback with regular falsi is that convergence may be one-sided. The happens in the situation illustrated in the next figure. Here $f(x)$ is concave upward and increasing; hence the secant is always above. As a result, w always lies to the left of the root. So, the bracketing interval only shrinks slowly (it is always of the form $[a_i, b_0]$). If $f(x)$ were concave downward and decreasing, w would always lie to the right of the root.



3 Modified Regula Falsi



One trick for avoiding the previous situation is simply to repeatedly decrease the "apparent value of $|f|$" at the unchanging endpoint, until the root estimate switches sides, as shown on the left.

In pseudo-code, we have

$F \leftarrow f(a_0)$
$G \leftarrow f(b_0)$
$w_0 \leftarrow a_0$
for $i = 0, 1, 2, \ldots,$ until satisfied, **do**
  $w_{i+1} \leftarrow (Ga_i - Fb_i)/(G - F)$
  if $f(a_i)f(w_{i+1}) \leq 0$
    then $a_{i+1} \leftarrow a_i$
      $b_{i+1} \leftarrow w_{i+1}$
      $G \leftarrow f(w_{i+1})$
      if $f(w_i)f(w_{i+1}) > 0$
        then $F \leftarrow F/2$
    else $a_{i+1} \leftarrow w_{i+1}$
      $b_{i+1} \leftarrow b_i$
      $F \leftarrow f(w_{i+1})$
      if $f(w_i)f(w_{i+1}) > 0$
        then $G \leftarrow G/2$

We run the modified regula falsi method on Example 1 and the results are as follows. Note the slightly faster convergence than with bisection (6 vs. 15 steps).

EXAMPLE 2. $f(x) = x^3 - x - 1$ (with modified regula falsi).

i	$[a_i, b_i]$	F	G	w_{i+1}	$f(w_{i+1})$
0	[1, 2]	-1	5	1.1667	-0.5787
1	[1.1667, 2]	-0.5787	2.5	1.3233	-0.0060
2	[1.3233, 2]	-0.0060	1.25	1.3265	-0.0078
3	[1.3233, 1.3265]	-0.0060	0.0078	1.3247	$-1.0221 \cdot 10^{-5}$
4	[1.3247, 1.3265]	$-1.0221 \cdot 10^{-5}$	0.0078	1.3247	$-1.7362 \cdot 10^{-8}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
6	[1.3247, 1.3247]	$-1.7362 \cdot 10^{-8}$	$1.7303 \cdot 10^{-8}$	1.3247	$2.2205 \cdot 10^{-16}$

Unlike bisection (which always halves the interval), root bracketing of the modified regula falsi method may not give a small interval of convergence. In general, a numerical routine terminates on one of the following conditions:

(a) $x_{i+1} - x_i$ is "small";

(b) $|f(x_i)|$ is "small";

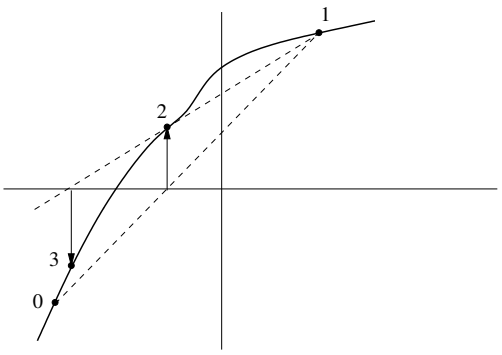
(c) i is "large".

One may wish to measure (a) and (b) as relative errors, say, respectively as

$$\begin{array}{lrcl} \text{(a)} & |x_{i+1} - x_i| & \leq & \text{XTOL} \cdot |x_i|, \\ \text{(b)} & |f(x_i)| & \leq & \text{FTOL} \cdot F, \end{array}$$

where XTOL and FTOL are some preset "tolerances" and F is an estimate of the magnitude.

4 Secant Method



The method starts with two estimates x_0 and x_1 and iterates as follows:

$$x_{i+1} = \frac{f(x_i)x_{i-1} - f(x_{i-1})x_i}{f(x_i) - f(x_{i-1})}. \tag{4}$$

Another very popular modification of the regular falsi is the *secant method*. It retains the use of secants throughout, but gives up the bracketing of the root. The secant method locates quite rapidly a point at which $|f(x)|$ is small but gives no general sense for how far away from a root of $f(x)$ this point might be. Also, $f(x_i)$ and $f(x_{i-1})$ need not be of opposite sign, so that the iteration formula (4) is prone to round-off errors. In an extreme situation, we might even have $f(x_i) = f(x_{i-1})$, making the calculation of x_{i+1} impossible. Although this does not cure the trouble, it is often better to calculate x_{i+1} from the equivalent expression

$$x_{i+1} = x_i - f(x_i)\frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})},$$

in which x_{i+1} is obtained from x_i by adding the "correction term"

$$-\frac{f(x_i)}{\Big(f(x_i) - f(x_{i-1})\Big)/(x_i - x_{i-1})}.$$

EXAMPLE 3. $f(x) = x^3 - x - 1$ (with secant method)

i	x_i	$f(x_i)$
0	1	-1
1	2	5
2	1.16666	-0.5787
3	1.253112	-0.28536
4	1.337206	0.05388
5	1.32385	-0.003698
6	1.3247079	$-4.27 \cdot 10^{-5}$
7	1.3247179	$3.458 \cdot 10^{-8}$

5 Newton's Method

In the secant method, we can write

$$x_{i+1} = x_i - \frac{f(x_i)}{f[x_i, x_{i-1}]},$$

where $f[x_i, x_{i-1}] = (f(x_i) - f(x_{i-1}))/(x_i - x_{i-1})$ is a first order divided difference that approximates the first derivative of f. Analogously, in the continuous case, the above suggests the formula

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}.$$

This is Newton's method. Essentially, x_{i+1} is the abscissa of the point where the x-axis intersects the line through $(x_i, f(x_i))$ with the slope $f'(x_i)$. It requires the knowledge of the derivative f'.

EXAMPLE 4. We now run Newton's method to find the unique real root of $f(x) = x^3 - x - 1$ using $f'(x) = 3x^2 - 1$.

i	x_i	$f(x_i)$
0	1	-1
1	1.5	0.875
2	1.347826	0.10068
3	1.325200	0.002058
4	1.324718	$9.2 \cdot 10^{-7}$
5	1.324718	$1.86 \cdot 10^{-13}$

i	x_i	$f(x_i)$
0	2	5
1	1.54545	1.14576
2	1.359615	0.1537
3	1.325801	0.00462
4	1.324718	$4.65 \cdot 10^{-6}$
5	1.324718	$4.7 \cdot 10^{-12}$

Newton's method converges if f is well-behaved and if the initial guess is near the root. Below we look at an example where Newton's method actually diverges.

EXAMPLE 5. Let $f(x) = \arctan x$. Then $x = 0$ is a solution of $f(x) = 0$. The Newton's iteration is defined by

$$x_{k+1} = x_k - (1 + x_k^2) \arctan x.$$

If we choose x_0 so that

$$\arctan |x_0| > \frac{2|x_0|}{1 + x_0^2},$$

then the sequence $|x_k|$ diverges, that is, $\lim_{k\to\infty} |x_k| = \infty$. The following diagram plots $\arctan x$, and marks two roots of the function $\arctan x - 2x/(1 + x^2)$, at which Newton's iteration will always yield each other. The chosen value x_0 is outside the interval ending at the two roots.

To see the divergence caused by the chosen x_0, consider the function $g(x) = x - (1 + x^2) \arctan x$. We have

$$\begin{aligned} g'(x) &= 1 - 1 - 2x \arctan x \\ &= -2x \arctan x \\ &< 0, \qquad \text{for all } x \neq 0. \end{aligned}$$

Since $g(0) = 0$, the above implies that $g(x) > 0$ when $x < 0$ and $g(x) < 0$ when $x > 0$. Subsequently,

$$x_{k+1} \cdot x_k = g(x_k) \cdot x_k < 0$$

and

$$\begin{aligned} \frac{|x_{k+1}|}{|x_k|} &= -\frac{x_{k+1}}{x_k} \\ &= \frac{-x_k + (1 + x_k^2) \arctan x_k}{x_k} \\ &= \frac{-|x_k| + (1 + |x_k|^2) \arctan |x_k|}{|x_k|}, \end{aligned}$$

when $x_k \neq 0$. Differentiating the function

$$h(x) = \frac{-x + (1+x^2)\arctan x}{x}$$

yields

$$h'(x) = \frac{(x - \arctan x) + x^2 \arctan x}{x}.$$

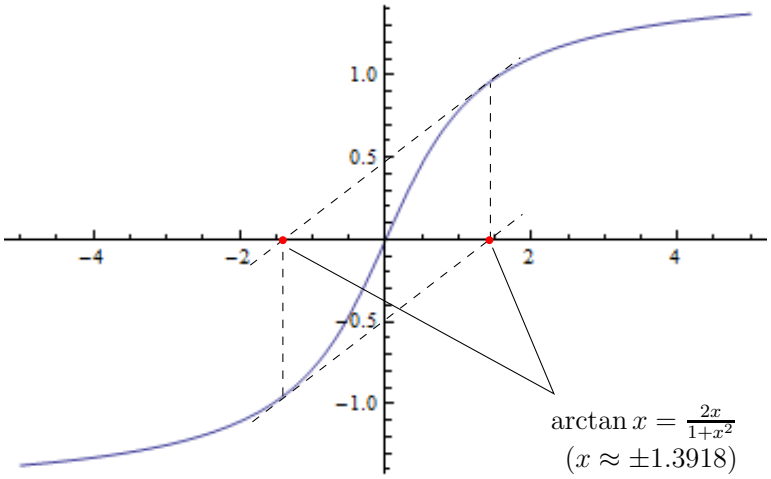
Note that $x > \arctan x > 0$ when $x > 0$. So $h'(x) > 0$ and $h(x)$ increase monotonically. Also we have $h(x_0) > 1$. Now we can easily show by induction:

$$\begin{aligned} |x_{k+1}| &> |x_k|, \\ h(|x_k|) &= \frac{|x_{k+1}|}{|x_k|} > \frac{|x_k|}{|x_{k-1}|} = h(|x_{k-1}|). \end{aligned}$$

Finally, we have

$$|x_k| = \frac{|x_k|}{|x_{k-1}|} \cdots \frac{|x_1|}{|x_0|} \cdot |x_0| > \left(\frac{|x_1|}{|x_0|}\right)^k \cdot |x_0|.$$

Thus the sequence $\{|x_k|\}$ diverges.



From the above example, we see that convergence for Newton's method is not guaranteed. For instance, if f' is near zero, the method can shoot off to infinity. Under what conditions will the method guarantee to converge?

Theorem 1 (Applicability of Newton's Method) *Let f be a twice continuously differentiable function over $[a, b]$. Suppose f satisfies the following conditions:*

i) $f(a)f(b) < 0$.

ii) $f'(x) \neq 0$ *for all* $x \in [a, b]$.

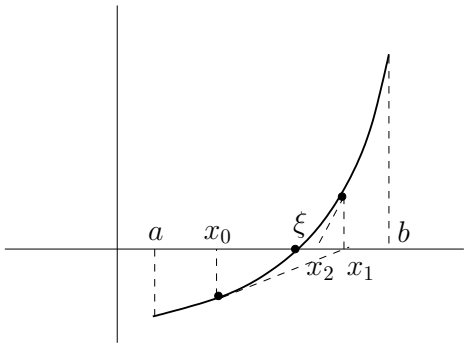
iii) $f''(x)$ *is either non-negative everywhere on* $[a, b]$ *or non-positive everywhere on* $[a, b]$.

iv)

$$\left|\frac{f(a)}{f'(a)}\right| < b-a \qquad \textit{and} \qquad \left|\frac{f(b)}{f'(b)}\right| < b-a.$$

Then Newton's method converges to the unique root of $f(x)$ in $[a,b]$ for any initial guess $x_0 \in [a,b]$.

Conditions i) and ii) ensure that there is exactly one zero in $[a,b]$. Condition iii) implies that f is either concave from above or concave from below. So conditions ii) and iii) together ensure that f' is monotone on $[a,b]$. Finally, condition iv) says that the tangent to the curve of f at either endpoint intersects the x-axis within the interval $[a,b]$. In the example on the right, $f''(x) \geq 0$, $f(a) < 0$, and $f(b) > 0$. The true root is at ξ. Observe that $x_1 > \xi$ always holds. And $x_k > \xi$ for all $k > 1$ and decrease monotonically to ξ.



6 Solving a System of Equations

It is generally difficult to solve several simultaneous non-linear equations. See [2, pp. 383-397]. However, suppose one has a "pretty good" neighborhood estimate of a simultaneous root. Then the higher-dimensional analogue of Newton's method is useful.

Suppose we have a sufficiently smooth mapping $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$, written as

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_n(x_1, \ldots, x_n) \end{pmatrix}.$$

For small $\Delta\boldsymbol{x} = (\Delta x_1, \ldots, \Delta x_n)^\top$ we can expand each f_i, $1 \leq i \leq n$, into Taylor series:

$$f_i(\boldsymbol{x} + \Delta\boldsymbol{x}) \;=\; f_i(\boldsymbol{x}) + \frac{1}{k!}\sum_{k=1}^{\infty}\left(\Delta x_1\frac{\partial}{\partial x_1} + \cdots + \Delta x_n\frac{\partial}{\partial x_n}\right)^k f_i\Big|_x$$

Given our current estimate $\boldsymbol{x}_k$ of a root of $\boldsymbol{f}$, we wish to computer $\Delta\boldsymbol{x}$ such that $\boldsymbol{x}_k + \Delta\boldsymbol{x}$ is a root of $\boldsymbol{f}$. Then

$$0 = f_i(\boldsymbol{x}_k) + \sum_{j=1}^{n}\frac{\partial f_i}{\partial x_j}\Delta x_j + \cdots, \qquad \text{for } i = 1, \ldots, n.$$

We ignore all terms of order 2 and above in Δx_j, $j = 1, \ldots, n$, and combine the n equations:

$$0 \approx \boldsymbol{f}(\boldsymbol{x}_k) = -J(\boldsymbol{x}_k)\Delta\boldsymbol{x}, \tag{5}$$

where

$$J(\boldsymbol{x}_k) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

is the Jacobian of $\boldsymbol{f}$ at $\boldsymbol{x} = \boldsymbol{x}_k$. Solve (5) for $\Delta\boldsymbol{x}$ to set up the iteration formula:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - J^{-1}(\boldsymbol{x}_k)\boldsymbol{f}(\boldsymbol{x}_k).$$

References

[1] M. Erdmann. Lecture notes for *16-811 Mathematical Fundamentals for Robotics*. The Robotics Institute, Carnegie Mellon University, 1998.

[2] W. H. Press, *et al*. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 2002.

[3] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag New York, Inc., 2nd edition, 1993.