CMPSC/Math 451, Numerical Computation

Wen Shen

Department of Mathematics, Penn State University

Problem: Given $f(x)$: continuous, real-valued, possibly non-linear.
Find a root r of $f(x)$ such that $f(r) = 0$.

Example 1. Quadratic polynomials: $f(x) = x^2 + 5x + 6$.

$$f(x) = (x+2)(x+3) = 0, \quad \Rightarrow \quad r_1 = -2, \quad r_2 = -3.$$

Observation: Roots are not unique.

Example 2. $f(x) = x^2 + 4x + 10 = (x+2)^2 + 6$. No roots.
Observation: There are no real r that would satisfy $f(r) = 0$.

Example 3. $f(x) = x^2 + \cos x + e^x + \sqrt{x+1}$.
Observation: Roots can be difficult/impossible to find analytically.

Overview of the chapter:

- Bisection
- Fixed point iteration
- Newton's method
- Secant method

Bisection method

Basic idea:

Given $f(x)$, a continuous function.
If we find some a and b, such that $f(a)$ and $f(b)$ are of opposite sign, then, there exists a point c, between a and b, such that $f(c) = 0$.

This fact follows from the continuity of f.

We can iterate on this idea!

Procedure:

- Initialization: Find a, b such that $f(a) \cdot f(b) < 0$.
  This means there is a root $r \in (a, b)$ s.t. $f(r) = 0$.
- Let $c = \frac{a+b}{2}$, mid-point.
- If $f(c) = 0$, done (lucky!)

  else:

  if $f(c) \cdot f(a) < 0$, pick the interval $[a, c]$
  if $f(c) \cdot f(b) < 0$, pick the interval $[c, b]$,
- Iterate the procedure until stop criteria satisfied.

Stop Criteria:

1) interval small enough, i.e., $(b - a) \leq \epsilon$,
2) $|f(c)|$ very small, i.e, $|f(c)| \leq \epsilon$
3) max number of iteration reached. (to avoid dead loop, in case the method does not converge.)
4) any combination of the previous ones.

Convergence analysis:

Consider $[a_0, b_0]$, $c_0 = \frac{a_0+b_0}{2}$, let $r \in (a_0, b_0)$ be a root.

$$\text{The error:} \qquad e_0 = |r - c_0| \le \frac{b_0 - a_0}{2}$$

Denote the further intervals as $[a_n, b_n]$ for iteration number n.

$$e_n = |r - c_n| \le \frac{b_n - a_n}{2} \le \frac{b_0 - a_0}{2^{n+1}} = \frac{e_0}{2^n}.$$

If the error tolerance is ε, we require $e_n \le \varepsilon$, then

$$\frac{b_0 - a_0}{2^{n+1}} \le \varepsilon \quad \Rightarrow \quad n \ge \frac{\ln(b-a) - \ln(2\varepsilon)}{\ln 2}, \quad (\# \text{ of steps})$$

Remark: very slow convergence.

Fixed point iterations

We rewrite the equation $f(x) = 0$ into the form $x = g(x)$.

Remark: This can always be achieved, for example: $x = f(x) + x$.
However, the choice of g makes a difference in convergence.

Main idea:
Make a guess of the solution, say $\bar{x}$. If the function $g(x)$ is "nice", then hopefully, $g(\bar{x})$ should be closer to the answer than $\bar{x}$. If that is the case, then we can iterate.

Iteration algorithm:

- Choose a start point x_0,
- Do the iteration $x_{k+1} = g(x_k)$, $k = 0, 1, 2, \cdots$ until meeting stop criteria.

Stop Criteria: Let ε be the tolerance

- $|x_k - x_{k-1}| \leq \varepsilon$,
- max # of iteration reached,
- any combination.

Example 1. Find an approximate solution to $f(x) = x - \cos x = 0$, with 4 digits accuracy.

Choose $g(x) = \cos x$, we have $x = \cos x$.
Choose $x_0 = 1$, and do the iteration $x_{k+1} = \cos(x_k)$:

$$\begin{aligned} x_1 &= \cos x_0 = 0.5403 \\ x_2 &= \cos x_1 = 0.8576 \\ x_3 &= \cos x_2 = 0.6543 \\ &\vdots \\ x_{23} &= \cos x_{22} = 0.7390 \\ x_{24} &= \cos x_{23} = 0.7391 \\ x_{25} &= \cos x_{24} = 0.7391 \quad \text{stop here} \end{aligned}$$

Our approximation to the root is 0.7391.

Example 2. Consider $f(x) = e^{-2x}(x-1) = 0$. (root: $r = 1$).

Rewrite as

$$x = g(x) = e^{-2x}(x-1) + x$$

Choose an initial guess $x_0 = 0.99$, very close to the real root. Iterations:

$$\begin{aligned} x_1 &= g(x_0) = 0.9886 \\ x_2 &= g(x_1) = 0.9870 \\ x_3 &= g(x_2) = 0.9852 \\ &\vdots \\ x_{27} &= 0.1655 \\ x_{28} &= -0.4338 \\ x_{29} &= -3.8477 \quad \text{Diverges. It does not work.} \end{aligned}$$

What went wrong?

Fixed point iteration; Convergence analysis.

Our iteration is $x_{k+1} = g(x_k)$. Let r be the exact root, s.t., $r = g(r)$. Define the error: $e_k = |x_k - r|$.

$$\begin{aligned} e_{k+1} &= |x_{k+1} - r| = |g(x_k) - r| = |g(x_k) - g(r)| \\ &= \left|g'(\xi)\right| |(x_k - r)| \qquad (\xi \in (x_k, r), \text{ since } g \text{ is continuous}) \\ &= \left|g'(\xi)\right| e_k \end{aligned}$$

$$\Rightarrow \quad e_{k+1} = \left|g'(\xi)\right| e_k.$$

Observation:

- If $|g'(\xi)| < 1$, then $e_{k+1} < e_k$, error decreases, the iteration convergence. (linear convergence)
- If $|g'(\xi)| > 1$, then $e_{k+1} > e_k$, error increases, the iteration diverges.

Convergence condition:

There exists an interval around r, say $[r-a, r+a]$ for some $a > 0$, such that $|g'(x)| < 1$ for almost all $x \in [r-a, r+a]$, and the initial guess x_0 lies in this interval.

In Example 1, $g(x) = \cos x$, $g'(x) = \sin x$, $r = 0.7391$,

$$\left|g'(r)\right| = |\sin(0.7391)| < 1. \qquad \text{OK, convergence.}$$

In Example 2, we have

$$\begin{aligned} g(x) &= e^{-2x}(x-1) + x, \\ g'(x) &= -2e^{-2x}(x-1) + x^{-2x} + 1 \end{aligned}$$

With $r = 1$, we have

$$\left|g'(r)\right| = e^{-2} + 1 > 1, \qquad \text{Divergence.}$$

Pseudo code

```
r=fixedpoint('g', x,tol,nmax)
  r=g(x); % first iteration
  nit=1;
  while (abs(r-g(r))>tol and nit < nmax) do
    r=g(r);
    nit=nit+1;
  end
end
```

A practical error estimate

Assume $|g'(x)| \leq m < 1$ in $[r-a, r+a]$. We have $e_{k+1} \leq m e_k$.
This gives

$$e_1 \leq m e_0, \quad e_2 \leq m e_1 \leq m^2 e_0, \quad \cdots \quad e_k \leq m^k e_0.$$

This result is useless unless we find a way to estimating e_0.

$$e_0 = |r - x_0| = |r - x_1 + x_1 - x_0| \leq e_1 + |x_1 - x_0| \leq m e_0 + |x_1 - x_0|$$

then

$$e_0 \leq \frac{1}{1-m} |x_1 - x_0|, \qquad \text{(can be computed)}$$

Put together

$$e_k \leq \frac{m^k}{1-m} |x_1 - x_0|.$$

$$e_k \le \frac{m^k}{1-m}\left|x_1 - x_0\right|.$$

If the error tolerance is ε, then

$$\frac{m^k}{1-m}\left|x_1 - x_0\right| \le \varepsilon,$$

$$m^k \le \frac{\varepsilon(1-m)}{\left|x_1 - x_0\right|}$$

$$k \ge \frac{\ln(\varepsilon(1-m)) - \ln\left|x_1 - x_0\right|}{\ln m}$$

which give the minimum number of iterations needed to achieve an error $\le \varepsilon$.

Example 3. We want to solve $\cos x - x = 0$ with the fixed point iteration

$$x = g(x) = \cos x, \qquad x_0 = 1,$$

with error tolerance $\varepsilon = 10^{-5}$. Find the min# iterations.
We know $r \in [0, 1]$.
We see that the iteration happens between $x = 0$ and $x = 1$.
For $x \in [0, 1]$, we have

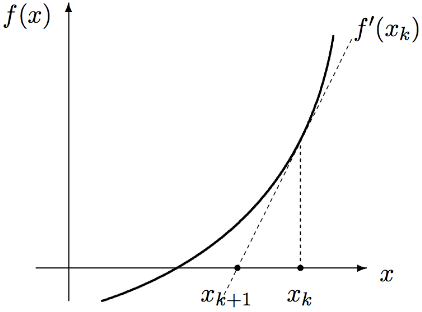
$$\left|g'(x)\right| = |\sin x| \leq \sin 1 = 0.8415, \qquad m \doteq 0.8415$$

And $x_1 = \cos x_0 = 0.5403$. Using the formula

$$k \geq \frac{\ln(\varepsilon(1-m)) - \ln|x_1 - x_0|}{\ln m} \approx 73 \quad \#\text{iterations needed}$$

In actual simulation $k = 25$ is enough.

Goal: Given $f(x)$, find a root r s.t. $f(r) = 0$.
Main idea: Given x_k, the next approximation x_{k+1} is determined by approximating $f(x)$ as a linear function at x_k.



We have

$$\frac{f(x_k)}{x_k - x_{k+1}} = f'(x_k)$$

which gives

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Newton's method is a fixed point iteration.

Newton's method is a fixed point iteration

$$x_{k+1} = g(x_k), \qquad g(x) = x - \frac{f(x)}{f'(x)}.$$

If r is a fixed point, assume $f'(r) \neq 0$, then

$$r = g(r), \quad r = r - \frac{f(r)}{f'(r)}, \quad \frac{f(r)}{f'(r)} = 0, \qquad f(r) = 0.$$

then r is a root for f.

Newton's method is the "best" fixed point iteration.

Observation: A fixed point iteration $x = g(x)$ is optimal if $g'(r) = 0$ where $r = g(r)$.

For Newton, we have

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

Then

$$g'(r) = \frac{f''(r)f(r)}{(f'(r))^2} = 0$$

which is the "best" possible scenario!

Newton's method; Convergence analysis.

Let r be the root so $f(r) = 0$ and $r = g(r)$. Recall $g'(r) = 0$.

$$\text{Define Error:} \qquad e_{k+1} = |x_{k+1} - r| = |g(x_k) - g(r)|$$

Taylor expansion for $g(x_k)$ at r:

$$\begin{aligned} g(x_k) &= g(r) + (x_k - r)g'(r) + \frac{1}{2}(x_k - r)^2 g''(\xi), \quad \xi \in (x_k, r) \\ &= g(r) + \frac{1}{2}(x_k - r)^2 g''(\xi) \end{aligned}$$

$$e_{k+1} = \frac{1}{2}(x_k - r)^2 \left|g''(\xi)\right| = \frac{1}{2} e_k^2 \left|g''(\xi)\right|$$

Write now $M = \frac{1}{2} \max_x |g''(\xi)|$, we have

$$e_{k+1} \le M\,(e_k)^2$$

This is called *quadratic convergence*.

Theorem: If $e_{k+1} \leq M\, e_k^2$, then $\lim_{k\to+\infty} e_k = 0$ if e_0 is sufficiently small. (This means, M can be big, but it would not effect the convergence!)
Proof for the convergence: We have

$$e_1 \leq (Me_0)e_0$$

If e_0 is small enough, such that $(Me_0) < 1$, then $e_1 < e_0$.
This means $(Me_1) < Me_0 < 1$, and so

$$e_2 \leq (Me_1)e_1 < e_1, \quad \Rightarrow \quad Me_2 < Me_1 < 1$$

By an induction argument, we conclude that $e_{k+1} < e_k$ for all k, i.e., error is strictly decreasing after each iteration. $\Rightarrow$ convergence.

Newton's iterations; Examples

Example . Find a numerical method to compute $\sqrt{a}$ using only $+, -, *, /$ arithmetic operations. Test it for $a = 3$.

Answer. It's easy to see that $\sqrt{a}$ is a root for $f(x) = x^2 - a$.
Newton's method gives

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - a}{2x_k} = \frac{x_k}{2} + \frac{a}{2x_k}$$

Test it on $a = 3$: Choose $x_0 = 1.7$.

	error
$x_0 = 1.7$	7.2×10^{-2}
$x_1 = 1.7324$	3.0×10^{-4}
$x_2 = 1.7321$	2.6×10^{-8}
$x_3 = 1.7321$	4.4×10^{-16}

Note the extremely fast convergence.

Sample Code:

```
r=newton('f','df',x0,nmax,tol)
x=x0;
n=0;
dx=f(x)/df(x);
while ((dx>tol) and (f(x)>tol)) or (n<nmax) do
  n=n+1;
  x=x-dx;
  dx=f(x)/df(x);
end
r=x-dx;
```

Secant method

If $f(x)$ is complicated, $f'(x)$ might not be available.
Solution for this situation: using approximation for f', i.e.,

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

This is *secant method*:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k)$$

Initial guess needed: x_0, x_1.

Advantages include

- No computation of f';
- One $f(x)$ computation each step;
- Also rapid convergence.

A bit on convergence: One can show that

$$e_{k+1} \le C e_k^{\alpha}, \qquad \alpha = \frac{1}{2}(1+\sqrt{5}) \approx 1.62$$

This is called *super linear convergence*. $(1 < \alpha < 2)$
It converges for all function f if x_0 and x_1 are close to the root r.

Example Use secant method for computing $\sqrt{a}$.
Answer. The iteration now becomes

$$x_{k+1} = x_k - \frac{(x_k^2 - a)(x_k - x_{k-1})}{(x_k^2 - a) - (x_{k-1}^2 - a)} = x_k - \frac{x_k^2 - a}{x_k + x_{k-1}}$$

Test with $a = 3$, with initial data $x_0 = 1.65$, $x_1 = 1.7$.

	error
$x_1 = 1.7$	7.2×10^{-2}
$x_2 = 1.7328$	7.9×10^{-4}
$x_3 = 1.7320$	7.3×10^{-6}
$x_4 = 1.7321$	1.7×10^{-9}
$x_5 = 1.7321$	3.6×10^{-15}

It is a little but slower than Newton's method, but not much.

The most practical algorithm: hybrid methods

- Sample the function to find a and b such that $f(a) \cdot f(b) < 0$.
- Use bisection's method, maybe 5-6 iterations, to get some good initial guess x_0;
- Use either Newton or secant method, with this x_0, for 3-4 iterations.