Newton's Method

Office hours schedule change
(till the end of semester)

TW 4:45 ~ 6:15 pm

Recap

Rootfinding

- Given : function f
- Want : scalar r such that

$$f(r) = 0.$$

Usually solved using iteration.

- Fixed-point iteration

  * $g(r) = r$ (intersection of $y = g(x)$ & $y = x$.)

  * Setting $g(x) = x - f(x)$,

  $\begin{pmatrix}\text{fixed point} \\ \text{of } g\end{pmatrix} = \begin{pmatrix}\text{root of} \\ f\end{pmatrix}$

  * $e_{k+1} \approx g'(r)\, e_k^{\boxed{1}}$

  FPI converges linearly.

Contents

1. Newton's Method (quadratic convergence ; fast !)
2. Secant Method (superlinear converge ; sort of fast, not as many comps.)
3. Appendix: Other Methods

Newton’s Method

Newton's Method

To find the root of f:

Newton's Method (Algorithm)

- Begin at the point $(x_0, f(x_0))$ on the curve and draw the tangent line at the point using the slope $f'(x_0)$:

$$y = f(x_0) + f'(x_0)(x - x_0).$$

- Find the x-intercept of the line and call it x_1:

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

- Continue this procedure to find $x_2, x_3, \ldots$ until the sequence converges to the root.

General iterative formula:

$$\boxed{x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}} \quad \text{for } k = 0, 1, 2, \ldots \qquad (\star)$$

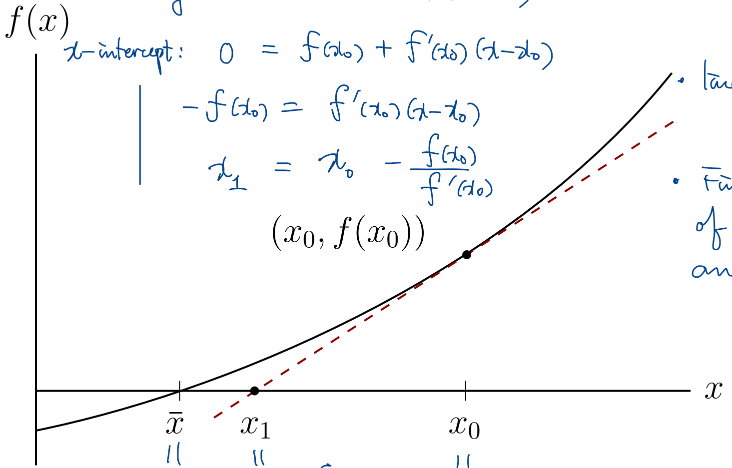
Newton's Method: Illustration

T-line: $y = f(x_0) + f'(x_0)(x - x_0)$

x-intercept: $0 = f(x_0) + f'(x_0)(x - x_0)$

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

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



- line tangent to $y = f(x)$ at $x = x_0$.
- Find the x-intercept of the tangent line and call it x_1.

$\bar{x} = r$

$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$, $f'(x_0) \neq 0$

x_0 = initial guess/iterate

Series Analysis

a root of f

Let $\epsilon_k = x_k - r$, $k = 1, 2, \ldots$, where r is the limit of the sequence and $f(r) = 0$.

Substituting $x_k = r + \epsilon_k$ into the iterative formula ($\star$):

$$\cancel{r} + \epsilon_{k+1} = \cancel{r} + \epsilon_k - \frac{f(r+\epsilon_k)}{f'(r+\epsilon_k)}$$

$x_{k+1} = r + \epsilon_{k+1}$

$$\epsilon_{k+1} = \epsilon_k - \frac{f(r+\epsilon_k)}{f'(r+\epsilon_k)}.$$

Taylor-expand f about $x = r$ and simplify (assuming $f'(r) \neq 0$):

intermediate steps shown below

$$\begin{aligned}
\epsilon_{k+1} &= \epsilon_k - \frac{\cancel{f(r)}^{\,0} + \epsilon_k f'(r) + \frac{1}{2}\epsilon_k^2 f''(r) + O(\epsilon_k^3)}{f'(r) + \epsilon_k f''(r) + O(\epsilon_k^2)} \\
&= \epsilon_k - \epsilon_k \left[1 + \frac{1}{2}\frac{f''(r)}{f'(r)}\epsilon_k + O(\epsilon_k^2)\right]\left[1 + \frac{f''(r)}{f'(r)}\epsilon_k + O(\epsilon_k^2)\right]^{-1} \\
&= \frac{1}{2}\frac{f''(r)}{f'(r)}\epsilon_k^2 + O(\epsilon_k^3).
\end{aligned}$$

$$\epsilon_{k+1} \approx \frac{1}{2}\frac{f''(r)}{f'(r)}\epsilon_k^{\boxed{2}}$$

(quad. convergence!)

$$\epsilon_{k+1} = \epsilon_k - \frac{f'(r)\,\epsilon_k + \frac{1}{2}f''(r)\,\epsilon_k^2 + O(\epsilon_k^3)}{f'(r) + f''(r)\,\epsilon_k + O(\epsilon_k^2)}$$

$$= \epsilon_k - \frac{\cancel{f'(r)}\,\epsilon_k\left[1 + \frac{1}{2}\frac{f''(r)}{f'(r)}\epsilon_k + O(\epsilon_k^2)\right]}{\cancel{f'(r)}\left[1 + \frac{f''(r)}{f'(r)}\epsilon_k + O(\epsilon_k^2)\right]}$$

Recall Geometric series

Let $\alpha \in \mathbb{R}$ such that $|\alpha| < 1$. Then

$$\sum_{k=0}^{\infty} \alpha^k = 1 + \alpha + \alpha^2 + \cdots$$

$$= \frac{1}{1-\alpha}.$$

$$= \epsilon_k - \epsilon_k\left[1 + \frac{1}{2}\frac{f''(r)}{f'(r)}\epsilon_k + O(\epsilon_k^2)\right]\left[1 - \frac{f''(r)}{f'(r)}\epsilon_k + O(\epsilon_k^2)\right]$$

$$= \epsilon_k - \epsilon_k\left[1 - \left(1 - \frac{1}{2}\right)\frac{f''(r)}{f'(r)}\epsilon_k + O(\epsilon_k^2)\right]$$

$$= \frac{1}{2}\frac{f''(r)}{f'(r)}\epsilon_k^2 + O(\epsilon_k^3)$$

Quad. Conv.

E.g.

k	e_k
4	10^{-2}
5	10^{-4}
6	10^{-8}
7	10^{-16}

$$(10^{-2})^2 = 10^{-4}$$

$$(10^{-4})^2 = 10^{-8}$$

$$\vdots$$

Series Analysis (cont')

- Previous calculation shows that $\epsilon_{k+1} \approx C\epsilon_k^2$, eventually. Written differently,

$$|\epsilon_{k+1}| \,/\, |\epsilon_k|^2 \to \text{(some positive number)}, \text{ as } k \to \infty.$$

  that is, each Newton iteration roughly squares the previous error. This is **quadratic convergence**.

- Alternately, note that

$$\log|\epsilon_{k+1}| \approx 2\log|\epsilon_k| + \text{(constant)},$$

  ignoring high-order terms. This means that the number of accurate digits[1] approximately doubles at each iteration.

[1] We say that an iterate is **correct within** p **decimal places** if the error is less than 0.5×10^{-p}.

Convergence of Newton's Method

r is a simple root of f.

Theorem 1 (Quadratic Convergence of Newton's Method)

Let f be twice continuously differentiable and $f(r) = 0$. If $f'(r) \neq 0$, then Newton's method is locally and quadratically convergent to r. The error $\epsilon_k = x_k - r$ at step k satisfies

$$\lim_{k\to\infty} \frac{|\epsilon_{k+1}|}{|\epsilon_k|^2} = \left| \frac{f''(r)}{2f'(r)} \right|.$$

quad.

Implementation

```
function x = newton(f,dfdx,x1)
% NEWTON   Newton's method for a scalar equation.
% Input:
%   f        objective function
%   dfdx     derivative function
%   x1       initial root approximation
% Output
%   x        vector of root approximations (last one is best)

% Operating parameters.
    funtol = 100*eps;  xtol = 100*eps;  maxiter = 40;

    x = x1;
    y = f(x1);
    dx = Inf;    % for initial pass below
    k = 1;

    while (abs(dx) > xtol) && (abs(y) > funtol) && (k < maxiter)
        dydx = dfdx(x(k));
        dx = -y/dydx;    % Newton step
        x(k+1) = x(k) + dx;

        k = k+1;
        y = f(x(k));
    end

    if k==maxiter, warning('Maximum number of iterations reached.'), end
end
```

→ residual → $|x_{k+1} - x_k|$ → max. # of iterations.

→ $f'(x_k)$

→ $-\dfrac{f(x_k)}{f'(x_k)}$

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

Note: Stopping Criteria

For a set tolerance, $\mathtt{TOL}$, some example stopping criteria are:

- Absolute error:

$$\left|x_{k+1} - x_k\right| < \mathtt{TOL}.$$

- Relative error: (useful when the solution is not too close to zero)

$$\frac{\left|x_{k+1} - x_k\right|}{\left|x_{k+1}\right|} < \mathtt{TOL}.$$

- Hybrid:

$$\frac{\left|x_{k+1} - x_k\right|}{\max(\left|x_{k+1}\right|, \theta)} < \mathtt{TOL},$$

  for some $\theta > 0$.

- Residual:

$$\left|f(x_k)\right| < \mathtt{TOL}.$$

Also useful to set a limit on the maximum number of iterations in case convergence fails.

Secant Method

Secant Method

- Newton's method requires calculation and evaluation of $f'(x)$, which may be challenging at times.
- The most common alternative to such situations is the **secant method**.
- The secant method replaces the instantaneous slope in Newton's method by the average slope using the last two iterates.

Secant Method (Algorithm)

- Begin with two initial iterates x_{-1} and x_0; draw the secant line connecting $(x_{-1}, f(x_{-1}))$ and $(x_0, f(x_0))$:

$f'(x_0)$

$$y = f(x_0) + \frac{f(x_0) - f(x_{-1})}{x_0 - x_{-1}}(x - x_0).$$

- Find the x-intercept of the line and call it x_1:

$$x_1 = x_0 - f(x_0)\frac{x_0 - x_{-1}}{f(x_0) - f(x_{-1})}.$$

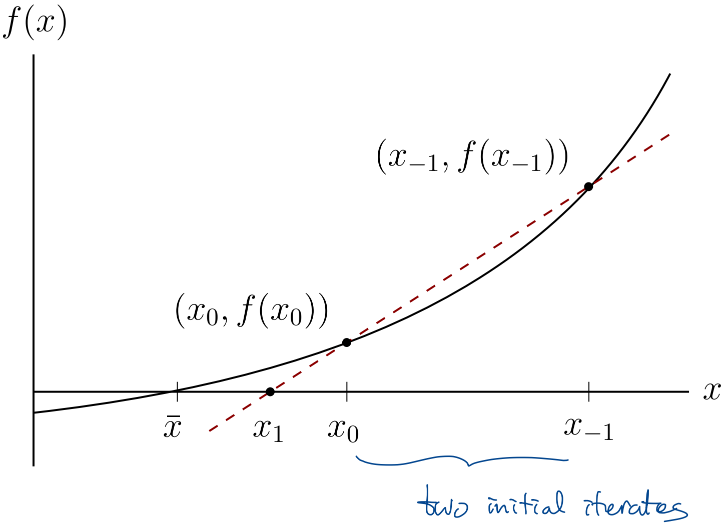
- Continue this procedure to find $x_2, x_3, \ldots$ until convergence is obtained.

General iterative formula:

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

$$\boxed{x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}} \quad \text{for } k = 0, 1, 2, \ldots$$

Secant Method: Illustration



Series Analysis

Assume that the secant method converges to r and $f'(r) \neq 0$. Let $\epsilon_k = x_k - r$ as before.

It can be shown that

$$\left|\epsilon_{k+1}\right| \approx \left|\frac{f''(r)}{2f'(r)}\right| \left|\epsilon_k\right| \left|\epsilon_{k-1}\right|,$$

which implies that

$$\boxed{\left|\epsilon_{k+1}\right| \approx \left|\frac{f''(r)}{2f'(r)}\right|^{\alpha-1} \left|\epsilon_k\right|^{\alpha},}$$

where

$$\alpha = \frac{1+\sqrt{5}}{2} \approx 1.618,$$

the *golden ratio*.

Therefore, the convergence of the secant method is **superlinear**; it lies between linearly and quadratically convergent methods.

Exercise. Confirm the statements in the previous page. Namely, show that

1. The error ϵ_k satisfies the approximate equation

$$|\epsilon_{k+1}| \approx \left|\frac{f''(r)}{2f'(r)}\right| |\epsilon_k| \, |\epsilon_{k-1}| \,.$$

2. If in addition $\lim_{k\to\infty} |\epsilon_{k+1}| / |\epsilon_k|^{\alpha}$ exists and is nonzero for some $\alpha > 0$, then

$$|\epsilon_{k+1}| \approx \left|\frac{f''(r)}{2f'(r)}\right|^{\alpha-1} |\epsilon_k|^{\alpha}, \quad \text{where } \alpha = \frac{1+\sqrt{5}}{2}.$$

Implementation

```
function x = secant(f,x1,x2)
% SECANT   Secant method for a scalar equation.
% Input:
%   f         objective function
%   x1,x2     initial root approximations
% Output
%   x         vector of root approximations (last is best)

% Operating parameters.
    funtol = 100*eps;  xtol = 100*eps;  maxiter = 40;

    x = [x1 x2];
    dx = Inf;  y1 = f(x1);
    k = 2;  y2 = 100;

    while (abs(dx) > xtol) && (abs(y2) > funtol) && (k < maxiter)
        y2 = f(x(k));
        dx = -y2 * (x(k)-x(k-1)) / (y2-y1);   % secant step
        x(k+1) = x(k) + dx;

        k = k+1;
        y1 = y2;    % current f-value becomes the old one next time
    end

    if k==maxiter, warning('Maximum number of iterations reached.'), end
end
```

Appendix: Other Methods

Inverse Interpolation

The **inverse quadratic interpolation** (IQI) is a generalization of the secant method to parabolas.

- Instead of using two most recent points (to determine a straight line), use three and obtain an quadratic interpolant.
- The parabola of the form $y = p(x)$ may have zero, one, or two x-intercept(s). So use the form $x = p(y)$, a parabola open sideways.

Algorithm.

- Begin with three initial iterates x_{-2}, x_{-1}, x_0; find the parabola of the form $x = p(y)$ passing through the three points $(x_{-2}, f(x_{-2}))$, $(x_{-1}, f(x_{-1}))$, and $(x_0, f(x_0))$.
- Find the x-intercept of the parabola and call it x_1.
- Continue the procedure to find $x_2, x_3, \ldots$ until convergence is obtained.

Inverse Interpolation (cont')

General iterative formula:

$$x_{k+1} = x_k - \frac{r(r-q)(x_k - x_{k-1}) + (1-r)s(x_k - x_{k-2})}{(q-1)(r-1)(s-1)}, \quad \text{for } k = 0, 1, 2, \ldots,$$

where

$$q = \frac{f(x_{k-2})}{f(x_{k-1})}, \quad r = \frac{f(x_k)}{f(x_{k-1})}, \quad s = \frac{f(x_k)}{f(x_{k-2})}.$$

Rather than deriving and implementing the formula, try using `polyfit` to perform the interpolation step.

Bisection Method: Bracketing a Root

The following is a corollary to the intermediate value theorem.

Theorem 2 (Existence of a Root)

Let f be a continuous function on $[a, b]$, satisfying $f(a)f(b) < 0$. Then f has a root between a and b, that is, there exists a number $r \in (a, b)$ such that $f(r) = 0$.

Bisection Method (cont')

Algorithm.

- Start with an interval $[a, b]$ where $f(a)f(b) \leqslant 0$.
- Bisect the interval into $[a, m] \cup [m, b]$ where $m = \; (a + b)/2$ is the midpoint.
- Select the subinterval in which $f(x)$ changes signs, ***i.e.***, calculate $f(a)f(m)$ and $f(m)f(b)$, choose the nonpositive one, and update the values of a and b.
- Repeat the process until you get close enough to the solution.

Notes

Let $[a, b]$ be the initial interval and let $[a_k, b_k]$ be the interval after k bisection steps.

- The length of $[a_k, b_k]$ is $(b-a)/2^k$.
- Using the midpoint $x_k = (a_k + b_k)/2$ as an estimate of the root r, note that

$$|\epsilon_k| = |x_k - r| < \frac{b-a}{2^{k+1}}.$$

- This accuracy is obtained by $k+2$ function evaluations.

Bisection Method: Pseudocode

```
while <a NOT CLOSE ENOUGH TO b>
  m = (a + b)/2;
  fm = f(m);
  if sign(fa) ~= sign(fm)
    b = m;
    fb = fm;
  else
    a = m;
    fa = fm;
  end
end
x_zero = .5*(a + b);
```