Lecture 40: Root Finding via the Secant Method

Newton's method is fast if one has a good initial guess x_0. Even then, it can be inconvenient (or impossible) and expensive to compute the derivatives $f'(x_k)$ at each iteration. The final root finding algorithm we consider is the *secant method*, a kind of *quasi-Newton method* based on an approximation of f'. It can be thought of as a hybrid between Newton's method and *regula falsi*.

7.3. Secant Method.

Throughout this semester, we saw how derivatives can be approximated using finite differences, for example,

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

for some small h. (Recall that too-small h will give a bogus answer due to rounding errors, so some caution is needed.) What if we replace $f'(x_k)$ in Newton's method with this sort of approximation? The natural algorithm that emerges is the *secant method*,

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

Note the similarity between this last formula and the *regula falsi* iteration:

$$c_k = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}.$$

Both methods approximate f by a line that joins two points on the graph of $f(x)$, but the secant method require no initial bracket for the root. Instead, the user simply provides *two* starting points x_0 and x_1 with no stipulation about the signs of $f(x_0)$ and $f(x_1)$. As a consequence, there is no guarantee that the method will converge: a poor initial guess can lead to divergence!

Do we recover the convergence behavior of Newton's method? Not quite, but the secant method (under suitable hypotheses) is *superlinear*, i.e., it is pth-order convergent with $p > 1$. In particular, it converges with order equal to the golden ratio, $\phi = \frac{1}{2}(1 + \sqrt{5}) \approx 1.6180$:

$$|e_{k+1}| \le C|e_k|^{\phi},$$

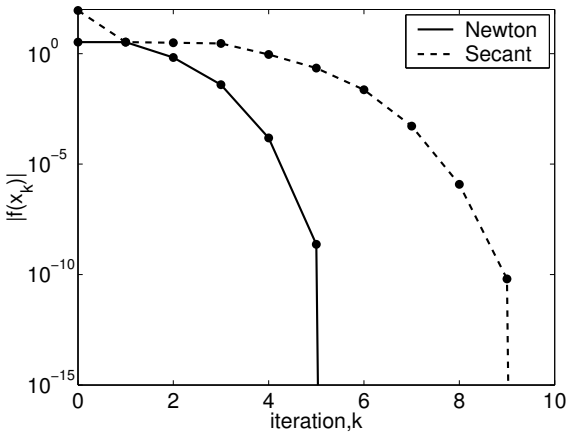
for a constant $C > 0$. Though you may regret that the secant method does not recover quadratic convergence, take solace in the fact that only one function evaluation $f(x_k)$ is required at each iteration, as opposed to Newton's method, which requires $f(x_k)$ and $f'(x_k)$. Typically the derivative is more expensive to compute than the function itself. If we assume that evaluating $f(x_k)$ and $f'(x_k)$ require the same amount of effort, then we can compute two secant iterates for roughly the same cost as a single Newton iterate. Two steps of the secant method combine to give an improved convergence rate:

$$|e_{k+2}| \;\le\; C|e_{k+1}|^{\phi} \;\le\; C\Big|C|e_k|^{\phi}\Big|^{\phi} \;\le\; C^{1+\phi}|e_k|^{\phi^2},$$

where $\phi^2 = \frac{1}{2}(3 + \sqrt{5}) \approx 2.62 > 2$. Hence, in terms of computing time, the secant method can actually be more efficient than Newton's method.[†]

The following plot shows the convergence of Newton's method on $f(x) = 1/x - 10$ with $x_0 = .15$, and the secant method with $x_0 = .01$ and $x_1 = .15$.

[†]This discussion is drawn from Kincaid and Cheney, *Numerical Analysis*, 3rd ed., §3.3.



k	x_k (Newton)	|f(x_k)| (Newton)	x_k (secant)	|f(x_k)| (secant)
0	0.15000000000000	-3.3333333e+00	0.01000000000000	9.0000000e+01
1	0.07500000000000	3.3333333e+00	0.15000000000000	-3.3333333e+00
2	0.09375000000000	6.6666667e-01	0.14500000000000	-3.1034483e+00
3	0.09960937500000	3.9215686e-02	0.07750000000000	2.9032258e+00
4	0.09999847412109	1.5259022e-04	0.11012500000000	-9.1940976e-01
5	0.09999999997672	2.3283064e-09	0.10227812500000	-2.2273824e-01
6	0.10000000000000	0.0000000e+00	0.09976933984375	2.3119343e-02
7			0.10000525472668	-5.2544506e-04
8			0.10000001212056	-1.2120559e-06
9			0.09999999999936	6.3689498e-11
10			0.10000000000000	0.0000000e+00

```
function xstar = secant(f, x0, x1)
% function xstar = secant(f, x0, x1)
% Compute a root of the function f using the secant method
% f:      a function name
% x0:     the starting guess
% x1:     the second starting point (defaults to x0+1)
% Example: secant('sin',3), or secant('my_f',1,1.1)


if (nargin < 3), x1 = x0+1; end
maxit = 60;
xprev = x0; fxprev = feval(f,xprev);
xcur  = x1; fxcur  = feval(f,xcur);  k=1;       % initialize
fprintf(' %3d  %20.14f  %10.7e\n', 0, xprev, fxprev);
fprintf(' %3d  %20.14f  %10.7e\n', 1, xcur, fxcur);
while (abs(fxcur) > 1e-15) & (k < maxit)
   x = xcur - fxcur*(xcur-xprev)/(fxcur-fxprev);     % Secant method
   xprev = xcur; fxprev = fxcur;
   xcur  = x;    fxcur  = feval(f,xcur);
   k = k+1;
   fprintf(' %3d  %20.14f  %10.7e\n', k, xcur, fxcur);
end
xstar = x;
```

7.4. A Parting Example: Kepler's Equation.

We close by describing the most famous nonlinear equation, developed in the first two decades of the seventeenth century by Johannes Kepler to solve the two-body problem in celestial mechanics. Kepler determined that a satellite trajectory forms an ellipse with its primary body at a focus point. Let $e \in [0, 1)$ denote the eccentricity of the ellipse and a denote the semi-major axis length. Assume that the satellite makes its closest approach to the primary, called periapsis, at time $t = 0$. The critical question is: Where is the satellite at time $t > 0$? We could numerically integrate the differential equations of motion (e.g., using the Störmer–Verlet algorithm), but in this simple setting there is a more direct approach that avoids differential equations. We measure the satellite's location by the angle ν swept out by the satellite's position vector from $\nu = 0$ at periapsis ($t = 0$) through to $\nu = 2\pi$ when the satellite returns to periapsis, having completed one orbit. If τ denotes the length of time of this one orbit (the *period*), then $M(t) = M := 2\pi t/\tau \in [0, 2\pi)$ describes the proportion of the period elapsed since the satellite last passed periapsis. Kepler solved the two-body problem by finding a formula for ν at a given time, t, $\tan(\nu/2) = \sqrt{(1+e)/(1-e)}\tan(E/2)$, where $E \in [0, 2\pi)$ is the *eccentric anomaly*, the solution of

$$M = E - e \sin E.$$

This nonlinear equation for the unknown E is Kepler's Equation, an innocuous formula that has received tremendous study.[‡] Despite this scrutiny, it turns out that Kepler's Equation is perfectly suited to the algorithms we study here, and is routinely solved in a few milliseconds. To determine the value of E, simply find the zero of $f(E) = M - E + e \sin E$ for $E \in [0, 2\pi)$.

...

Kepler's equation is but one important and beautiful example of numerical analysis at its most effective: an ideal algorithm applied to a well-conditioned problem gives amazing accuracy in an instant.

I hope our investigations this semester have given you a taste of the beautiful mathematics that empower numerical computations, the discrimination to pick the right algorithm to suit your given problem, the insight to identify those problems that are inherently ill-conditioned, and the tenacity to always seek clever, efficient solutions.

[‡]See Peter Colwell, *Solving Kepler's Equation Over Three Centuries*, Willmann-Bell, 1993.