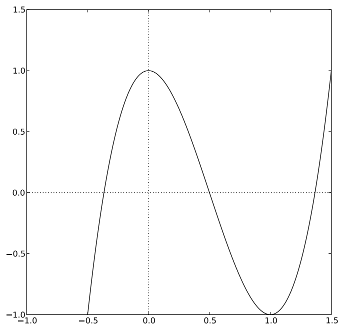
Scientific Programming in Python
Worksheet
25 September 2009

Root finding

We want to find the roots of $f(x) = 4x^3 - 6x^2 + 1$ using a numerical method. See the figure below for a plot of the function. For this function we happen to know that the true roots are at $\frac{1}{2}$ and $\frac{1 \pm \sqrt{3}}{2}$.



In general we would like to find the roots of any continuous function. We will explore three different methods for root finding—the bisection method, the secant method, and the Newton-Raphson method[1]. Your task is to implement each of the root finding methods described below.

The bisection method

The bisection method starts from an interval that is known to contain a root of the function. That is, we require as input a and b such that $\text{sign}(f(a)) \neq \text{sign}(f(b))$. The method then divides the interval into halves and selects the half that contains the root. So with $c = (a + b)/2$ we select the interval

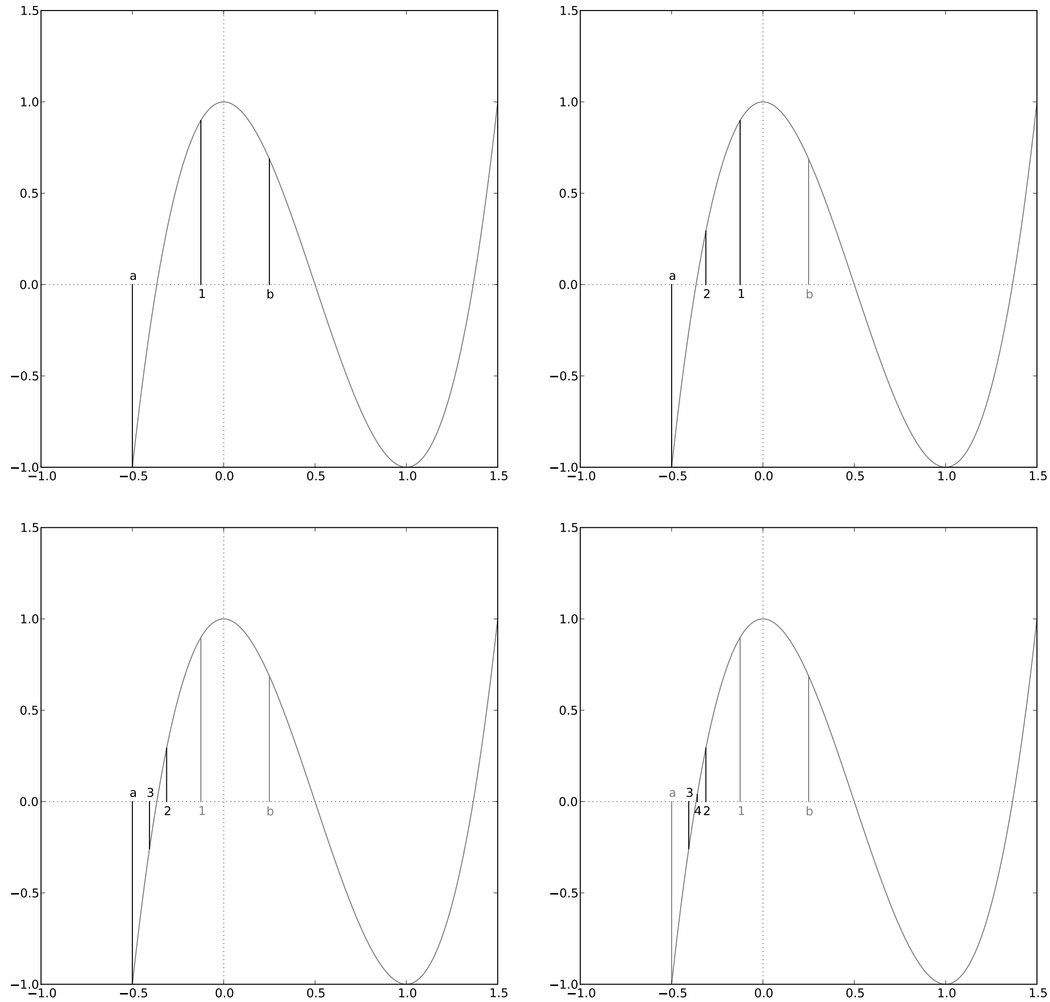
$$\begin{array}{ll} [a, c] & \text{if } \text{sign}(f(a)) \neq \text{sign}(f(c)), \text{ or} \\ [c, b] & \text{if } \text{sign}(f(b)) \neq \text{sign}(f(c)). \end{array}$$

The methods then repeats with this new interval. The image below shows what happens if we start from the interval $\left[-\frac{1}{2}, \frac{1}{4}\right]$.

- In the first step the function value at the center point is positive and we choose the left interval.
- In the second step the function value at the center of the new interval is also positive and we choose the left interval again.
- In the third step the function value at the center is negative and we choose the right interval.

With each step we divide the interval containing the root in half and get a better approximation to the true solution.

[1] You can use Wikipedia (`http://en.wikipedia.org/`) to find out more about the bisection, secant and Newton-Raphson methods.



Implement a function `bisection(f, a, b)` that finds a root of the function `f` in the interval defined by `a` and `b`. For example calling `bisection(f, -0.5, 0.25)` should give the result -0.366.

The secant method

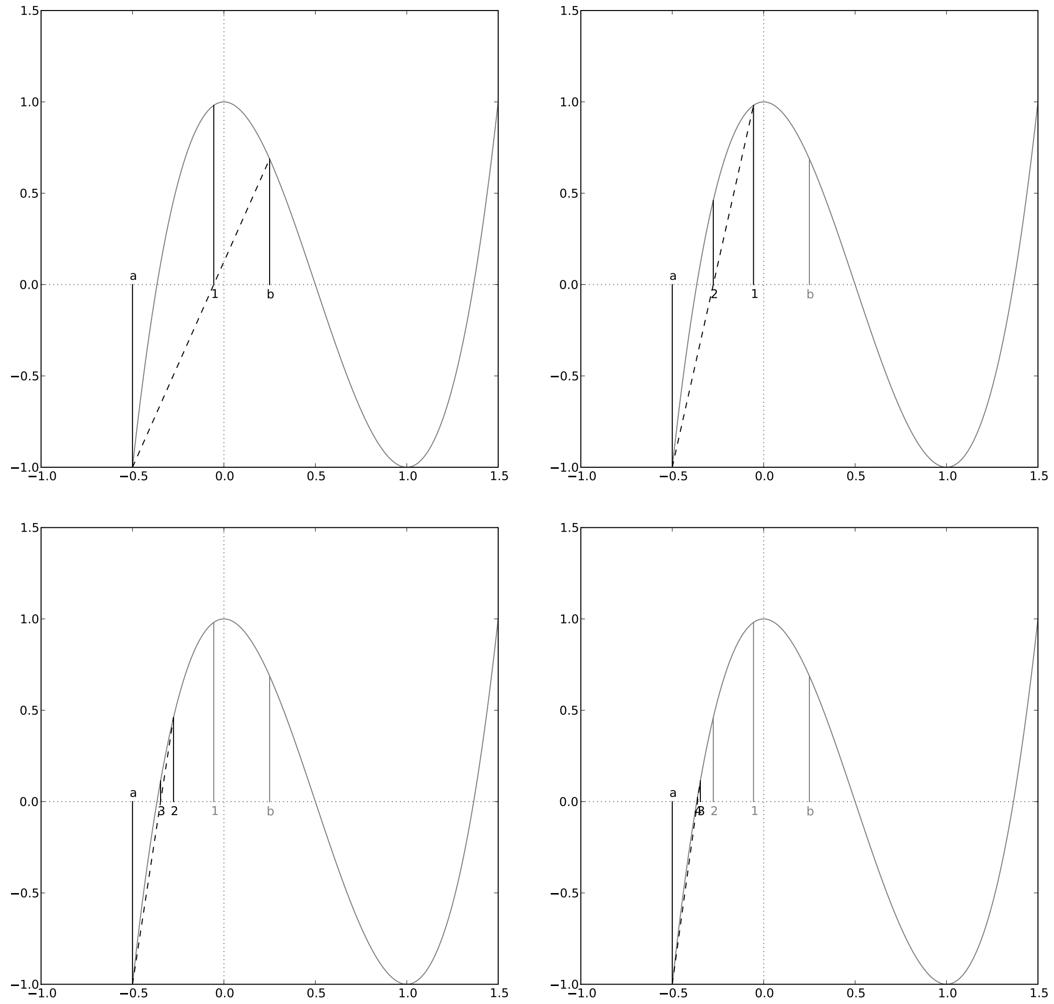
The secant method also starts from an interval, $[a, b]$ that contains a root of the function. Rather than dividing the interval in half, we divide it at the point where the line that joins $(a, f(a))$ and $(b, f(b))$ intersects the x-axis. To do this we set

$$c = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

and again select the new interval as

$$\begin{aligned} &[a, c] \qquad \text{if } \operatorname{sign}(f(a)) \neq \operatorname{sign}(f(c)), \text{ or} \\ &[c, b] \qquad \text{if } \operatorname{sign}(f(b)) \neq \operatorname{sign}(f(c)). \end{aligned}$$

The image below shows the first 3 steps of the secant method.



Implement a function `secant(f, a, b)` that finds a root of `f` in the interval defined by `a` and `b`.

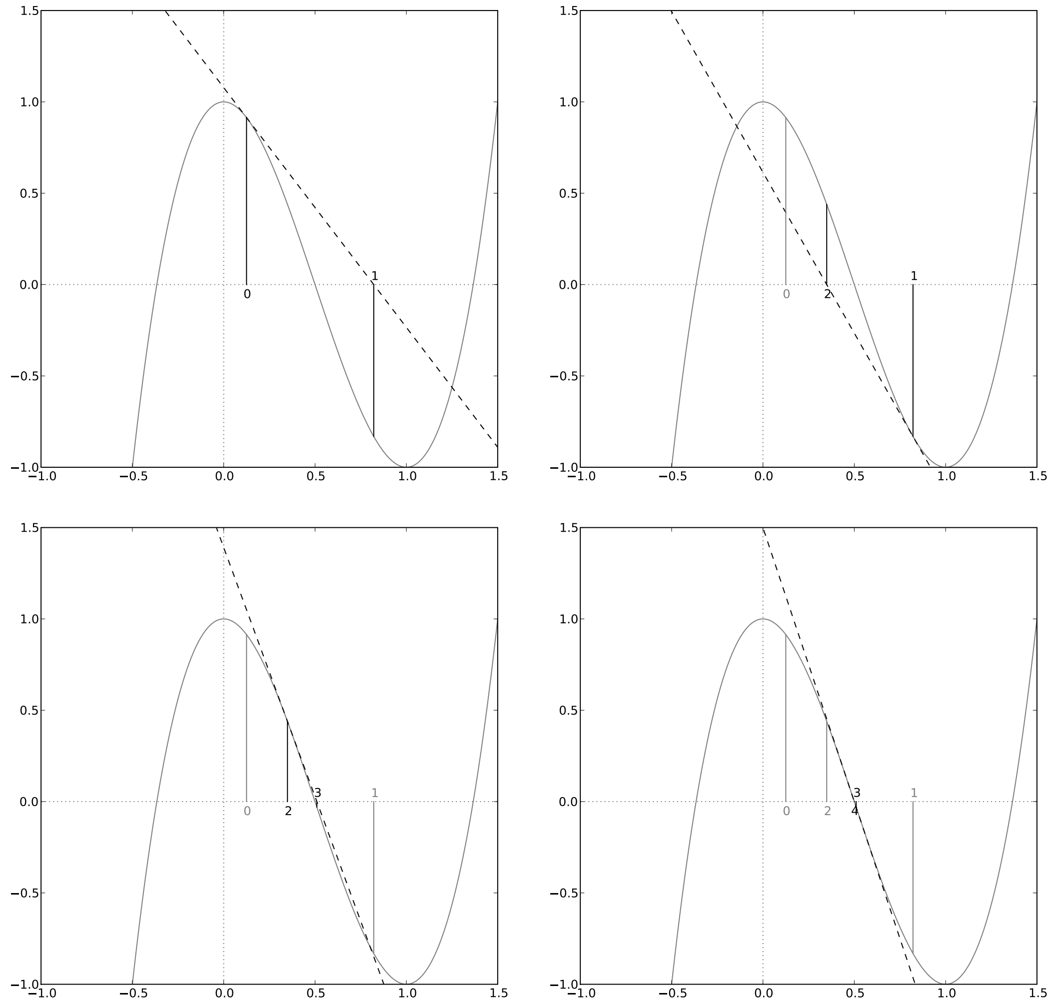
Newton's method

Newton's method—also called the Newton-Raphson method—does not require an initial interval that contains a root of the function. It requires only one point from which to start. This method does however assume that the derivative of the function, $f'(x)$ can be evaluated at any x.

From the current estimate of the root, x_i, the next point is found as the intersection between the x-axis and the line tangent to f at x_i. That is,

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

The figure below demonstrates the first three steps of Newton's method.



Implement a function `newton(f, g, x)` that finds a root of `f` using the gradient function `g` and the initial estimate `x`.

Final note

Which of the above methods is the best? Or does it depend on the function, $f(x)$? Experiment with how fast each method converges for different functions and different initial conditions. How many steps do you need before the error in the estimate of the root is less than some tolerance, ϵ?