1 Root Finding

Solutions to the equation $f(x) = 0$ in one dimension.

1.1 Bisection Method

Given an interval $[a, b]$ containing a root, the interval is interatively halved. The interval contains a root if $f(a)f(b) < 0$.

The Bisection method always converges, so long as a root exists on the inteval given. The bisection method converges linearly, making it one of the slower options.

```
function [p] = bisection(f, a, b, iterations)

    fa = f(a);
    p = a;
    fp = fa;

    for i=1:iterations

        % set p to center of the interval
        p = (a + b) / 2.0;
        fp = f(p);

        % test location of root in the interval
        if fp*fa > 0
            a = p;
            fa = fp;
        else
            b = p;
        end

        fprintf('Iteration %3.0d: p = %4.9f, f(p) = %4.9f \n', i, p, f(p));

    end
end
```

1.2 Fixed Point Iteration

A fixed point is the solution to $f(p) = p$. One is not guaranteed to exist.

Given a starting point p, a function $g(x) = x - f(x)$ is constructed, and its series is iterated.

$$x_i = g(x_{i-1})$$

The solution for p exists where the solution converges. Fixed Point Iteration converges if g is continuous on its range, and its range contains a fixed point.

```
function [p] = fixed_point(f, p, iterations)
    for i=1:iterations

        p = p - f(p);

        fprintf('Iteration %3.0d: p = %4.9f, f(p) = %4.9f \n', i, p, f(p));

    end
```

```
end
```

1.3 Newton's Method

Given a starting point p, and the function f', its series is iterated.

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

Fixed Point iteration is then applied. Newton's Method converges quadratically, but requires the derivative of the function and the initial guess to be close to the root.

```
function [p] = newton(f, fp, p, iterations)

    for i=1:iterations

        p = p - f(p)/fp(p);

        fprintf('Iteration %3.0d: p = %4.9f, f(p) = %4.9f \n', i, p, f(p));

    end
end
```

1.4 Aitken's Δ^2 Method

The Δ^k represents the k-order finite-derivative and is defined such that $\Delta p_n = p_{n+1} - p_n$ and $\Delta^k p_n = \Delta(\Delta^{k-1} p_n)$.

Using the first- and second-order finite-derivates, a series can be constructed.

$$\hat{x}_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}$$

This series can then be applied with fixed point iteration. Aitken's Δ^2 Method generally converges much faster than the original series.

1.5 Horner's Method

Given that function P is a polynomial with n real roots, and polynomial Q with no real roots, function P can be factorized by Q.

$$P(x) = Q(x) \prod_{i=0}^{n} (x - x_i) + a_0$$

Horner's Method is rapid for polynomials and can find all zeros through recursion. Programmatically, the polynomial is represented as a coefficient array.