PS 1

Due: Weds, Jan 28

1: By the book Book section 3.6, problems 1, 4, 5

3.6.1 Apply bisection to find the root of $f(x) = \sqrt{x} - 1.1$ starting from the interval $[0, 2]$ with $\texttt{atol} = 10^{-8}$.

1. How many iterations are required? Does the iteration count match the expectations?

   This takes 27 iterations, which is $\lceil \log_2 2 \cdot 10^8 \rceil - 1$, exactly as expected.

2. What is the resulting absolute error? Could this absolute error be predicted by the convergence analysis?

   The error is $-6.6 \cdot 10^{-9}$, which is less than 10^{-8} (as expected). Other than a bound on the error, the convergence analysis for bisection gives no information about the error magnitude.

3.6.4 Consider the function $g(x) = x^2 + 3/16$.

1. What are the fixed points? Solve $g(x) - x = 0$ or $x^2 - x + 3/16 = 0$; the roots of this quadratic are

$$x_- = \frac{1}{4}, \quad x_+ = \frac{3}{4}.$$

2. For $x = x_- + \delta$ where δ is small, we have

$$\begin{aligned} g(x) &= g(x_-) + g'(x_-)\delta + O(\delta^2) \\ &= x_- + 2x_-\delta \end{aligned}$$

   and similiarly for x_+. Because $|2x_-\delta| < |\delta|$ and $|2x_+\delta| > |\delta|$, respectively, the fixed point at x_- is attractive and the fixed point at x_+ is repulsive.

3. For the fixed point at x_-, the error iteration is $\delta_{k+1} = \frac{1}{2}\delta_k + O(\delta_k^2)$. Once δ_k is sufficiently small, it will take roughly $\log_2 10 \approx 3.3$ steps to cut the error by a factor of 10 (or, more precisely, about ten steps to cut the order by 10^3).

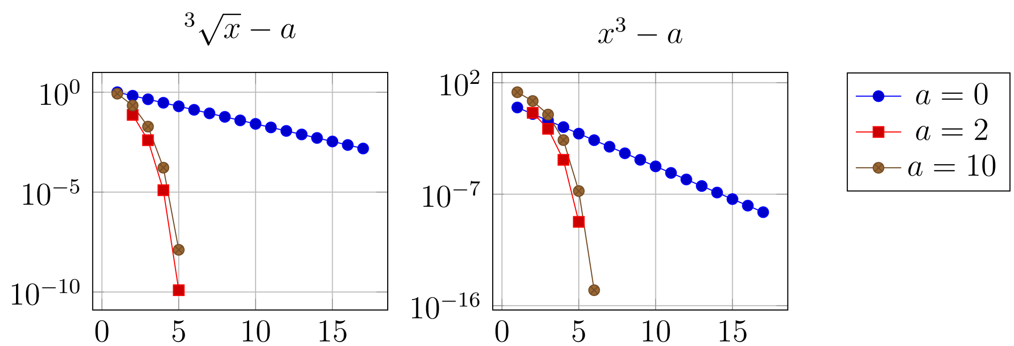

Figure 1: Error in x_k and residual error $x_k^3 - a$ for $a = 0$ (initial guess 1), $a = 2$ (initial guess 1), and $a = 10$ (initial guess 2).

3.6.5 We begin the Newton iteration with $x_0 = 1, 1, 2$ for $a = 0, 2, 10$. Starting the $a = 0$ iteration with $x_0 = 0$ is perhaps more natural, but does not illustrate the feature that the problem is supposed to illustrate (shown in Figure 1).

Note that in a "real" implementation of a cube root function (e.g. as part of a C or Fortran system math library), I would make explicit use of the floating point representation to get a guess: integer division for the exponent in order to reduce the problem to a fixed range – an method known as argument reduction – followed by a modest-degree polynomial approximation to get an initial guess. This is beyond the scope of the problem, but is worth having heard about.

The iterations for $\sqrt[3]{a}$ for $a = 2, 10$ both have the characteristic shape of quadratic convergence on a semi-logarithmic plot. The iteration for $a = 0$ is more interesting. In this case, the function $f(x) = x^3$ has a multiple root at the origin, and so Newton iteration converges only linearly. More precisely, the iteration for $a = 0$ is

$$x_{k+1} = \frac{2}{3} x_k,$$

and so the semilog error curve is $\log x_k = \log x_0 + k \log(2/3)$.

The code follows:

```
function [x, xs, fs] = ps1cubic(a,x)

  xs = [];
  fs = [];
```

```
f = x^3-a;
for k = 1:100
  xs = [xs, x];
  fs = [fs, f];
  if abs(f) < 1e-8
    return;
  end
  fp = 3*x^2;
  x = x-f/fp;
  f = x^3-a;
end
error('Did not converge after 100 iterations');
```

2: Water, water The dispersion relation for shallow water waves is

$$\omega^2 = k\left(g + \frac{T}{\rho}k^2\right)\tanh(kh)$$

where

$$\begin{aligned} h &= \text{water depth} \\ k &= \text{spatial wave number } (2\pi \text{ / wave length}) \\ \omega &= \text{frequency } (2\pi \text{ / period}) \\ T &= \text{surface tension} \\ \rho &= \text{mass density} \\ g &= \text{gravitational acceleration.} \end{aligned}$$

For water at 25C, $T/\rho = 7.2 \times 10^{-5}$ N/m^4, and the acceleration due to gravity is $g = 9.8$ m/s^2. Assuming these values, write a code using Newton's method to find k given ω and h, assuming $kh \ll 1$. Your routine should take the form

```
function k = ps2water(omega, h)
```

Answer: The only tricky thing is to find a good initial guess for Newton's method. The easiest way to do this is to use the small size of kh to approximate $\tanh(kh) \approx kh$ in order to get a quadratic in k^2. The positive root of the quadratic gives a very good starting guess, and Newton finishes the job.

```
function k = ps2water(w, h)
%
% Approximate wave length for shallow water equations
% Dispersion relation is
%   w^2 = (g * k + T/rho * k^3) * tanh(k*h);
% where
%   w = 2 pi / t = frequency   (t = period)
%   k = 2 pi / L = wave number (L = wave length)
%   h = bottom depth
% and the physical constants for water on earth (at 25C) are

g = 9.8;                % gravity
T_div_rho = 7.2e-5; % surface tension / mass

% Goal: find L
%
% Approximate tanh(k*h) = 2*k*h
%   w^2 = k^2 * (g + T/rho * k^2)*h
% This is a quadratic in k^2 of the form
%   a (k^2)^2 + b k^2 + c = 0
% where

w2 = w^2;
a = h*T_div_rho;
b = h*g;
c = -w2;

% We're looking for the positive root.  Use the stable formula.

k2 = 2*c/(-b-sqrt(b^2-4*a*c));
k  = sqrt(k2);

% Okay, now we need to refine. Let's use Newton. We'll stop
% after ten steps or after an error estimate of 10eps, whichever
% comes first.
```

```
for step = 1:10
  tanh_kh = tanh(k*h);
  f   = k * (g + T_div_rho * k^2) * tanh_kh-w2;
  df = (g + 3*T_div_rho*k) * tanh_kh + ...
        k * (g + T_div_rho*k^2) * (1-tanh_kh)*(1+tanh_kh)*h;
  dk = f/df;
  k = k-dk;
  if abs(dk/k) < 10*eps, break; end
end
```