Problem 1 The Fibonacci numbers f_n are defined by

$$f_{n+2} = f_{n+1} + f_n$$

with $f_0 = 0$ and $f_1 = 1$.

(a) Show that

$$\frac{f_{n+1}}{f_n} \to \varphi = \frac{1+\sqrt{5}}{2}$$

as $n \to \infty$.

(b) Determine the rate of convergence of f_{n+1}/f_n to φ.

Solution 1 (5 pts x 2 parts = 10 pts)

Define

$$R_n = \frac{f_{n+1}}{f_n}$$

for $n \geq 1$.

(a) The Fibonacci recursion gives us a corresponding recursion for R_n:

$$R_n = \frac{f_{n+1}}{f_n} = \frac{f_n + f_{n-1}}{f_n} = 1 + \frac{1}{R_{n-1}}.$$

With

$$\varphi = \frac{1+\sqrt{5}}{2} = 1 + \frac{1}{\varphi},$$

and using the fact that $R_n > 1$ for all $n \geq 2$,

$$\begin{aligned}
|R_n - \varphi| &= \left| \left(1 + \frac{1}{R_{n-1}}\right) - \left(1 + \frac{1}{\varphi}\right) \right| \\
&= \left| \frac{1}{R_{n-1}} - \frac{1}{\varphi} \right| \\
&= \left| \frac{R_{n-1} - \varphi}{R_{n-1}\varphi} \right| \\
&\leq \frac{1}{\varphi} |R_{n-1} - \varphi| \\
&\leq \left(\frac{1}{\varphi}\right)^{n-1} |R_1 - \varphi|.
\end{aligned}$$

Since $0 < 1/\varphi < 1$, we have

$$\lim_{n\to\infty} \left(\frac{1}{\varphi}\right)^{n-1} |R_1 - \varphi| = 0,$$

and thus

$$\lim_{n\to\infty} |R_n - \varphi| = 0.$$

Hence we get

$$\lim_{n\to\infty} \frac{f_{n+1}}{f_n} = \lim_{n\to\infty} R_n = \varphi.$$

(b) From part (a), we have

$$|R_n - \varphi| \le \left(\frac{1}{\varphi}\right)^{n-1} |R_1 - \varphi| = O\left(\frac{1}{\varphi^n}\right).$$

Therefore we get the rate of convergence

$$R_n = \varphi + O\left(\frac{1}{\varphi^n}\right).$$

In fact, since $R_{n-1} \to \varphi$, we can get the result

$$|R_n - \varphi| \le \frac{1}{R_{n-1}\varphi} |R_{n-1} - \varphi| = O\left(\frac{1}{\varphi^{2n}}\right).$$

We can also get this result by using the explicit formula

$$f_n = \frac{1}{\sqrt{5}} \left(\varphi^n - (1-\varphi)^n\right).$$

We then find that

$$|R_n - \varphi| = \left|\frac{(1-\varphi)^{n+1} - \varphi(1-\varphi)^n}{\varphi^n - (1-\varphi)^n}\right| = \left(\frac{\varphi - 1}{\varphi}\right)^n \left|\frac{1-2\varphi}{1 - \left(\frac{1-\varphi}{\varphi}\right)^n}\right| = O\left(\left(\frac{\varphi-1}{\varphi}\right)^n\right).$$

Since $\varphi - 1 = 1/\varphi$, we get the rate of convergence $R_n = \varphi + O(1/\varphi^{2n})$.

Problem 2 Consider the fixed point iteration

$$x_{n+1} = \frac{-x_n^2 - c}{2b}, \qquad (*)$$

where b and c are fixed real parameters.

(a) If $x_n \longrightarrow x$, what does x solve?

(b) Analyze and sketch the region of (b, c) values where $(*)$ converges at a rate $\mathcal{O}\left(2^{-n}\right)$ or better from an interval of starting values x_0 near x.

Solution 2 (5 pts x 2 parts = 10 pts)

(a) Taking limits on both sides of the given equation

$$x_{n+1} = -\frac{-x_n^2 - c}{2b}$$

gives

$$x = -\frac{x^2 + c}{2b}.$$

That is, x solves

$$x^2 + 2bx + c = 0.$$

(b) Define

$$g(x) = -\frac{x^2 + c}{2b}.$$

Then the derivative satisfies

$$|g'(x)| = \frac{|x|}{|b|} \leq \frac{1}{2}$$

whenever $|x| \leq \frac{|b|}{2}$.

Therefore in order to make the fixed point iteration converge at a rate $\mathcal{O}(2^{-n})$ or better, we need conditions on b and c such that $[-\frac{|b|}{2}, \frac{|b|}{2}]$ is invariant under the function g. Examining $g'(x)$ allows us to conclude that the extrema of g occur at $x = 0$ and $x = \pm b/2$, at which g evaluates to

$$g(0) = -\frac{c}{2b},$$
$$g(\pm b/2) = \frac{-b^2 - 4c}{8b}.$$

Case 1: $b > 0$

In order to have the invariant property, we need

$$-\frac{c}{2b} \leq \frac{b}{2},$$

and

$$\frac{-b^2-4c}{8b} \geq -\frac{b}{2}.$$

Straightforward computations give

$$-b^2 \leq c \leq \frac{3}{4}b^2.$$

Case 2: $b < 0$

In order to have the invariant property, we need

$$\frac{-b^2-4c}{8b} \leq -\frac{b}{2},$$

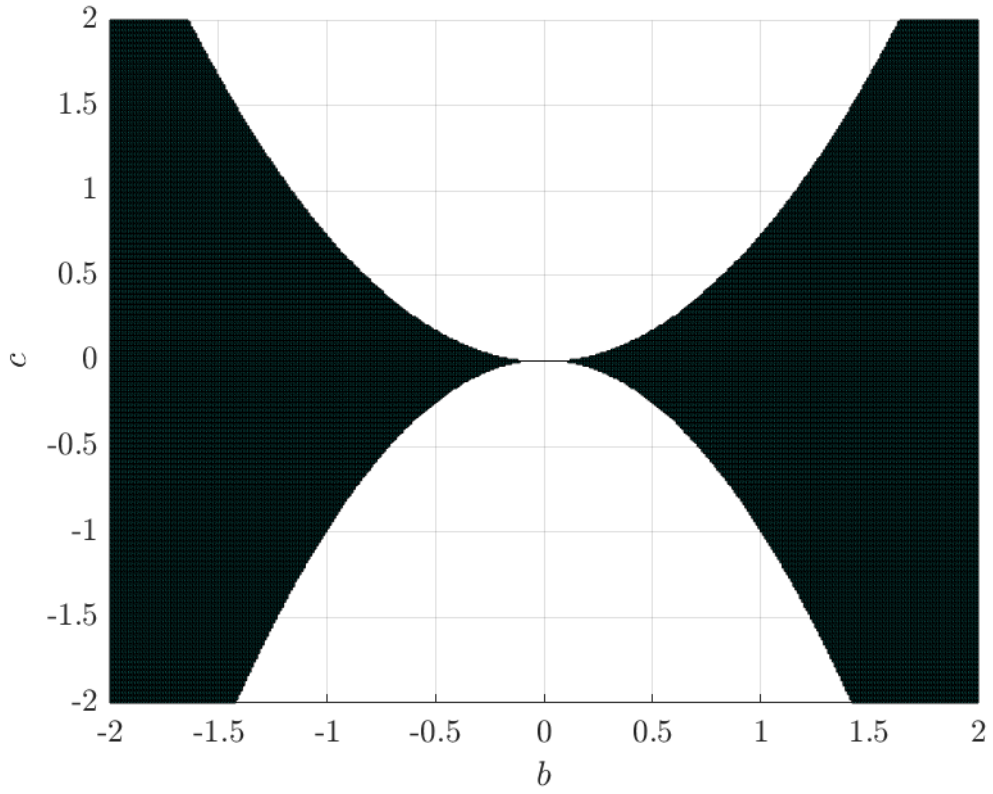
and

$$-\frac{c}{2b} \geq \frac{b}{2}.$$

Straightforward computations again give

$$-b^2 \leq c \leq \frac{3}{4}b^2.$$

The plot for the region $\left\{(b,c) \,|\, -b^2 \leq c \leq \frac{3}{4}b^2\right\}$ is as follows:



Problem 3 Consider the fixed point iteration

$$x_{n+1} = -b - \frac{c}{x_n} = g(x_n) \qquad (**)$$

(a) Show that $|g'(x)| \leq \frac{1}{2}$ whenever $x^2 \geq 2\,|c|$.

(b) Show that $g(x)^2 \geq 2\,|c|$ whenever $x^2 \geq 2\,|c|$ and $b^2 \geq \frac{9}{2}\,|c|$.

(c) Draw the region of (b, c)-space where $(**)$ converges at a rate at least $\mathcal{O}\,(2^{-n})$ from any starting point x_0 with $x_0^2 \geq 2|c|$.

Solution 3 (5 pts x 3 parts = 15 pts)

(a) If $c = 0$ then $g = -b$ and the conclusion clearly holds.

If $c \neq 0$ then

$$g(x) = -b - \frac{c}{x},$$

and we have

$$|g'(x)| = \frac{|c|}{x^2} \leq \frac{1}{2}$$

iff $x^2 \geq 2|c|$.

Whenever $x^2 \geq 2|c|$ and $b^2 \geq \frac{9}{2}|c|$, we have that

$$|g(x)| = \left|b + \frac{c}{x}\right| \geq |b| - \frac{|c|}{|x|} \geq \frac{3}{\sqrt{2}}\sqrt{|c|} - \frac{|c|}{\sqrt{2|c|}} = \sqrt{2|c|},$$

which implies that $g(x)^2 \geq 2|c|$.

(b) Case 1: $c = 0$.

Clearly this gives the desired property.

Case 2: $c \neq 0$ and $b^2 \geq \frac{9}{2}|c|$.

It can be seen that there is exactly one fixed point x^* (i.e. $x^* = -b - \frac{c}{x^*}$) in the set $A = (-\infty, -\sqrt{2|c|}] \cup [\sqrt{2|c|}, \infty)$. (For proof, see note at the end.)

Then we have

$$|x_{n+1} - x^*| = \left|\left(-b - \frac{c}{x_n}\right) - \left(-b - \frac{c}{x^*}\right)\right| = |c|\left|\frac{1}{x_n} - \frac{1}{x^*}\right| = \frac{|c|}{|x^*||x_n|}|x_n - x^*|$$

Since $x^* \in A$ we get $|x^*| \geq \sqrt{2|c|}$, and by part (b), if the starting point x_0 satisfies $x_0^2 \geq 2|c|$, we have $|x_n| \geq \sqrt{2|c|}$.

Plugging in these estimates yields

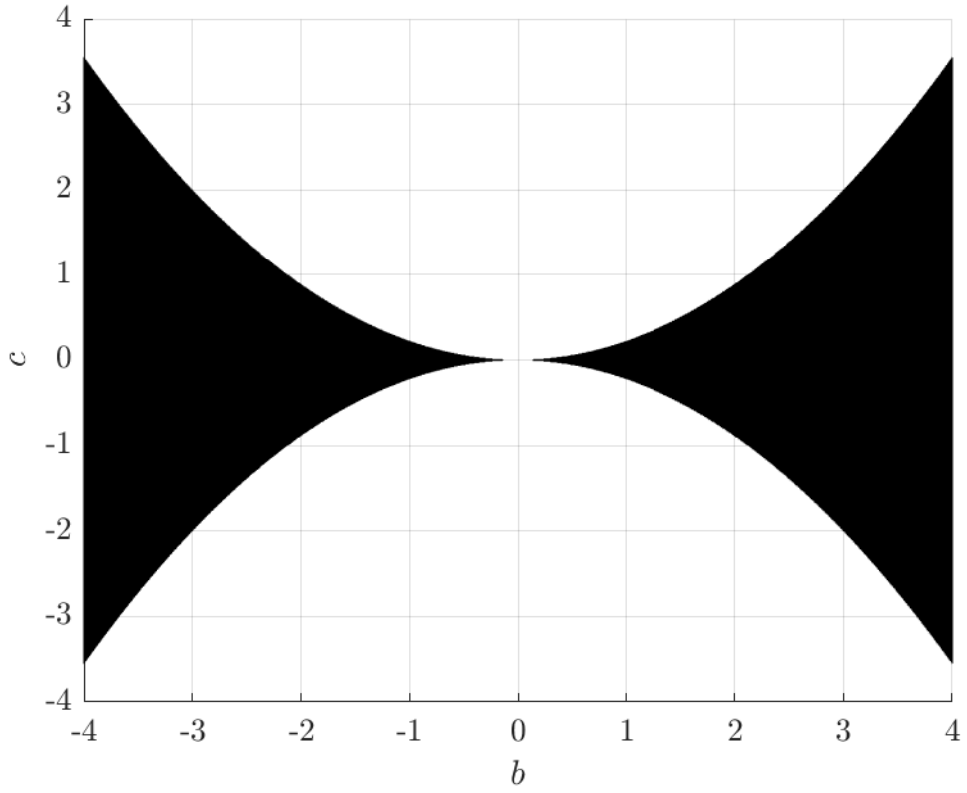
$$|x_{n+1} - x^*| \leq \frac{|c|}{|x^*||x_n|}|x_n - x^*| \leq \frac{1}{2}|x_n - x^*|.$$

We can apply it repeatedly to obtain

$$|x_n - x^*| \leq 2^{-n}|x_0 - x^*|,$$

that is, $x_n = x^* + \mathcal{O}(2^{-n})$.

Altogether, the region is $\{(b, c) \,|\, b^2 \geq \frac{9}{2}|c|\}$. The plot is given as follows:



Note:

The condition $b^2 \geq \frac{9}{2}|c|$ implies the discriminant for the quadratic equation $g(x) = x$ is

$$b^2 - 4c \geq \frac{9}{2}|c| - 4|c| = \frac{1}{2}|c| > 0.$$

Hence we have two real roots

$$x_\alpha = \frac{-b + \sqrt{b^2 - 4c}}{2}$$

and

$$x_\beta = \frac{-b - \sqrt{b^2 - 4c}}{2}.$$

Next we prove the existence and uniqueness of the root in the set A.

If they were both in A, we would have $|x_\alpha||x_\beta| \geq (\sqrt{2|c|})^2 = 2|c|$, which contradicts the fact that $x_\alpha x_\beta = c$. On the other hand, if $b \geq \frac{3}{\sqrt{2}}\sqrt{|c|}$, we have that

$$x_\beta = \frac{-b - \sqrt{b^2 - 4c}}{2} \leq \frac{-\frac{3}{\sqrt{2}}\sqrt{|c|} - \sqrt{(\frac{3}{\sqrt{2}}\sqrt{|c|})^2 - 4|c|}}{2} = -\sqrt{2|c|},$$

and if $b \leq -\frac{3}{\sqrt{2}}\sqrt{|c|}$, we have that

$$x_\alpha = \frac{-b + \sqrt{b^2 - 4c}}{2} \geq \frac{\frac{3}{\sqrt{2}}\sqrt{|c|} + \sqrt{(-\frac{3}{\sqrt{2}}\sqrt{|c|})^2 - 4|c|}}{2} = \sqrt{2|c|}.$$

Problem 4 Fix $a > 0$ and consider the fixed point iteration

$$x_{n+1} = x_n(2 - ax_n). \qquad (***)$$

(a) Show that if $x_n \to x$ then $x = \frac{1}{a}$ or $x = 0$.

(b) Find an interval (α, β) containing $\frac{1}{a}$ such that $(***)$ converges to $1/a$ whenever $x_0 \in (\alpha, \beta)$.

(c) Find the rate of convergence in (b).

Solution 4 (5 pts x 3 parts = 15 pts)

(a) Taking the limit on both sides of the given recursive relation

$$x_{n+1} = x_n(2 - ax_n)$$

gives

$$x = x(2 - ax),$$

which is

$$x(1 - ax) = 0.$$

We get $x = \frac{1}{a}$ or $x = 0$.

(b) Set $y_n = ax_n - 1$. The given recursive relation

$$x_{n+1} = x_n(2 - ax_n)$$

can be written as

$$y_{n+1} = ax_{n+1} - 1 = ax_n(2 - ax_n) - 1 = -(ax_n - 1)^2 = -y_n^2.$$

It follows that

$$|y_n| = |y_0|^{2^n}.$$

In order to have the convergence of the sequence $\{x_n\}_{n=0}^{\infty}$ we need that $|y_0| < 1$, i.e.

$$|ax_0 - 1| < 1,$$

which is

$$\left| x_0 - \frac{1}{a} \right| < \frac{1}{a}.$$

Hence taking $x_0 \in \left(0, \frac{2}{a}\right)$ gives

$$\lim_{n\to\infty} x_n = \frac{1}{a}.$$

Note that $x_0 = 0$ or $x_0 = \frac{2}{a}$ implies that $x_n = 0$ for any $n \geq 1$, and therefore that

$$\lim_{n\to\infty} x_n = 0.$$

Hence whenever $x_0 \in [\alpha, \beta]$ (with $\alpha = 0, \beta = \frac{2}{a}$) we have the convergence of $\{x_n\}_{n=0}^{\infty}$.

(c) Notice that

$$|ax_n - 1| = |ax_0 - 1|^{2^n}$$

from part (b) tells us that

$$\left| x_n - \frac{1}{a} \right| = \frac{1}{a} |ax_0 - 1|^{2^n},$$

which indicates that

$$x_n = \frac{1}{a} + O(|ax_0 - 1|^{2^n}).$$

This sequence therefore converges quadratically; one can compute that

$$\frac{\left| x_{n+1} - \frac{1}{a} \right|}{\left| x_n - \frac{1}{a} \right|^2} = a.$$

Problem 5 (a) Write down Newton's method in the form

$$x_{k+1} = g(x_k)$$

for solving

$$f(x) = x^2 - 2bx + b^2 - d^2 = 0$$

where $b > 0$ and $d > 0$ are parameters.

(b) Show that $|g'(x)| \leq 1/2$ whenever $|x - b| \geq d/\sqrt{2}$.

(c) Show that $|g(x) - b| \geq d/\sqrt{2}$ whenever $|x - b| \geq d/\sqrt{2}$.

(d) Sketch the graph of $f(x)$ with the roots of $f(x) = 0$ and the intervals of x where Newton's method is guaranteed to converge.

Solution 5 (5 pts x 4 parts = 20 pts)

(a) Using the fact that

$$g(x) = x - \frac{f(x)}{f'(x)},$$

we have

$$g(x) = x - \frac{x^2 - 2bx + b^2 - d^2}{2(x-b)} = \frac{x^2 - b^2 + d^2}{2(x-b)}.$$

(b) The quotient rule for differentiation gives

$$g'(x) = \frac{2x \cdot 2(x-b) - (x^2 - b^2 + d^2) \cdot 2}{4(x-b)^2} = \frac{(x-b)^2 - d^2}{2(x-b)^2} = \frac{1}{2} - \frac{d^2}{2(x-b)^2}.$$

We know $d^2/(x-b)^2 \geq 0$ and we are given $d^2/(x-b)^2 \leq 2$. Therefore we have

$$|g'(x)| \leq \frac{1}{2}.$$

(c) Notice that

$$\begin{aligned} g(x) - b &= \frac{x^2 - b^2 + d^2}{2(x-b)} - b \\ &= \frac{x+b}{2} + \frac{d^2}{2(x-b)} - b \\ &= \frac{x-b}{2} + \frac{d^2}{2(x-b)} \\ &= (x-b)\left(\frac{1 + \frac{d^2}{(x-b)^2}}{2}\right). \end{aligned}$$

The AM-GM inequality $|2xy| \leq x^2 + y^2$ gives

$$|g(x) - b| = |x - b| \left(\frac{1 + \frac{d^2}{(x-b)^2}}{2} \right) \geq |x - b| \frac{d}{|x - b|} = d \geq \frac{d}{\sqrt{2}}.$$

(d) We can solve

$$f(x) = x^2 - 2bx + b^2 - d^2 = 0$$

to get $x = b - d$ or $x = b + d$, and these two roots are actually the fixed points for g.

We are going to show the following statements for the convergence of the sequence $\{x_k\}_{k=0}^{\infty}$ generated by Newton's method and the initial approximation x_0.

(1.) For $x_0 \in (-\infty, b)$, the sequence $\{x_k\}_{k=0}^{\infty}$ converges to $b - d$. (2.) For $x_0 \in (b, \infty)$, the sequence $\{x_k\}_{k=0}^{\infty}$ converges to $b + d$.

Statement 1. If $x < b$ then $x - b < 0$, so the result from part (c) implies that $g(x) - b$ is negative and therefore

$$g(x) - b \leq -\frac{d}{\sqrt{2}}.$$

This implies that $x_n \in (-\infty, b - \frac{d}{\sqrt{2}}]$ for $n \geq 1$. In particular, we know that $(-\infty, b - \frac{d}{\sqrt{2}}]$ is invariant under g. By part (b), we have that

$$|g'(x)| \leq \frac{1}{2}$$

whenever $x \in (-\infty, b - \frac{d}{\sqrt{2}}]$. Therefore the fixed-point convergence theorem guarantees that $\{x_k\}_{k=0}^{\infty}$ converges to the unique fixed point $x = b - d$ for g in the interval $(-\infty, b - d/\sqrt{2}]$. □

Statement 2. If $x > b$ then $x - b > 0$, so the result from part (c) implies that $g(x) - b$ is positive and therefore

$$g(x) - b \geq \frac{d}{\sqrt{2}}.$$

This implies that $x_n \in [b + \frac{d}{\sqrt{2}}, -\infty)$ for $n \geq 1$. In particular, we know that $[b + \frac{d}{\sqrt{2}}, -\infty)$ is invariant under g. By part (b), we have that

$$|g'(x)| \leq \frac{1}{2}$$

whenever $x \in [b + \frac{d}{\sqrt{2}}, -\infty)$. Therefore the fixed-point convergence theorem guarantees that $\{x_k\}_{k=0}^{\infty}$ converges to the unique fixed point $x = b + d$ for g in the interval $[b + \frac{d}{\sqrt{2}}, -\infty)$. □

Altogether as long as $x_0 \in (b, \infty)$, the sequence $\{x_k\}_{k=0}^{\infty}$ converges to $b + d$.

To conclude, if $x_0 = b$, Newton's method is not defined. Other than that, Newton's method is guaranteed to converge.

Problem 6 Implement Newton's method in a Matlab or Octave program `newton.m` of the form

```
function r = newton(x0, f, p, n)
% x0: initial estimate of the root
% f: function and derivative handle [ y, yp ] = f(x, p)
% p: parameters to pass through to f
% n: number of steps
```

(a) Use `newton.m` to find an approximation to within ϵ to the first positive value of x with $x = 2\sin x$. Report the number of steps, the final result, and the absolute and relative errors. Characterize the convergence as linear or quadratic by tabulating the number of correct bits at each step of the iteration.

(b) Use `newton.m` as many times as necessary to find all solutions $x > 0$ of the equation

$$f(x) = \frac{1}{x} + \ln x - 2 = 0.$$

Report the number of steps, the final result, and the absolute and relative errors. Characterize the convergence as linear or quadratic by tabulating the number of correct bits at each step of the iteration.

(c) Use `newton.m` to solve the equation

$$f(x) = (x - \epsilon^3)^3 = 0$$

Report the number of steps, the final result, and the absolute and relative errors. Characterize the convergence as linear or quadratic by tabulating the number of correct bits at each step of the iteration. Explain your results.

(d) Use `newton.m` to solve the equation

$$f(x) = \arctan(x - \epsilon^2) = 0$$

for a diverse selection of starting values. Find starting values which lead to convergence, divergence and oscillation. Report the number of steps, the final result, and the absolute and relative errors.

Solution 6 (10 pts x 1 code + 5 pts x 4 parts = 30 pts)

Sample code follows:

```
1  function r = newton(x0, f, p, n)
2  %NEWTON Performs Newton's method
3  % x0: initial estimate of the root
4  % f: function and derivative handle [y, yp] = f(x, p)
5  % p: parameters to pass through to f
6  % n: number of steps
7
8      r = x0;
```

```
    for k = 1:n
        [y, yp] = f(r, p);
        r = r - y/yp;
    end

end
```

For the next few problems we use the following sample code to tabulate the number of correct bits at each step of the iteration and examine our results:

```
function newton_results(x0, f, p, n)

for k = 1 : n

    r = newton(x0, f, p, k);
    matlab_result = fzero( f, r);
    matlab_result = eps**2

    abs_err = abs(matlab_result - r);
    rel_err = abs_err/abs(matlab_result);
    bits = -log( abs_err/abs(matlab_result) ) / log( 2.0 );

  line = sprintf( ' steps = %d, r = %20.16g, rel err =%9.5g, bits=%9.5g ', k
     , r, rel_err, bits );
  disp( line )

  endfor

end
```

(a) We find the root as follows:

```
function [ y, yp ] = p6a( x, p )
 y = x - 2 * sin( x );
 yp = 1 - 2 * cos( x );
end

octave:1> newton_results(2, @p6a, 1, 10)
 steps = 1, r =    1.900995594203909, rel err =0.0029023, bits=   8.4286
 steps = 2, r =    1.895511645379595, rel err =9.1682e-06, bits=   16.735
 steps = 3, r =    1.895494267208713, rel err =9.2183e-11, bits=   33.337
 steps = 4, r =    1.895494267033981, rel err =        0, bits=      Inf
 steps = 5, r =    1.895494267033981, rel err =        0, bits=      Inf
 steps = 6, r =    1.895494267033981, rel err =        0, bits=      Inf
 steps = 7, r =    1.895494267033981, rel err =        0, bits=      Inf
```

```
 steps = 8, r =    1.895494267033981, rel err =        0, bits=      Inf
 steps = 9, r =    1.895494267033981, rel err =        0, bits=      Inf
 steps = 10, r =    1.895494267033981, rel err =        0, bits=      Inf
```

The number of correct bits slightly more than doubles at each iteration when close to the root, a demonstration of quadratic convergence.

(b) We find the two roots as follows:

```
function [ y, yp ] = p6b( x, p )
 y = 1./x + log(x) - 2;
 yp = -1./x.^2 + 1./x;
end

octave:1> newton_results(.2, @p6b, 1, 10)
 steps = 1, r =    0.269528104378295, rel err =  0.15201, bits=   2.7177
 steps = 2, r =   0.3092191411091564, rel err = 0.027137, bits=   5.2036
 steps = 3, r =   0.3175584625305333, rel err =0.00089972, bits=   10.118
 steps = 4, r =   0.3178441157620177, rel err =9.9778e-07, bits=   19.935
 steps = 5, r =   0.3178444328989826, rel err =1.2264e-12, bits=   39.569
 steps = 6, r =   0.3178444328993727, rel err =        0, bits=      Inf
 steps = 7, r =   0.3178444328993727, rel err =        0, bits=      Inf
 steps = 8, r =   0.3178444328993727, rel err =        0, bits=      Inf
 steps = 9, r =   0.3178444328993727, rel err =        0, bits=      Inf
 steps = 10, r =   0.3178444328993727, rel err =        0, bits=      Inf
octave:2> newton_results(5, @p6b, 1, 10)
 steps = 1, r =    6.191013047286873, rel err =  0.01814, bits=   5.7847
 steps = 2, r =    6.304550119075867, rel err =0.00013404, bits=   12.865
 steps = 3, r =    6.305395233304984, rel err =7.2901e-09, bits=   27.031
 steps = 4, r =    6.305395279271691, rel err =        0, bits=      Inf
 steps = 5, r =    6.305395279271691, rel err =        0, bits=      Inf
 steps = 6, r =    6.305395279271691, rel err =        0, bits=      Inf
 steps = 7, r =    6.305395279271691, rel err =        0, bits=      Inf
 steps = 8, r =    6.305395279271691, rel err =        0, bits=      Inf
 steps = 9, r =    6.305395279271691, rel err =        0, bits=      Inf
 steps = 10, r =    6.305395279271691, rel err =        0, bits=      Inf
```

The number of correct bits slightly more than doubles at each iteration when close to the root, a demonstration of quadratic convergence.

(c) We find the root as follows:

```
function [ y, yp ] = p6c( x, p )
 y = ( x - eps**3 ) ** 3;
 yp = 3 * ( x - eps**3 ) **2;
end

octave:8> newton_results(eps, @p6c, 1, 270)
```

```
 steps = 1, r = 1.480297366166876e-16, rel err =1.3522e+31, bits=  -103.42
 steps = 2, r = 9.86864910777917e-17, rel err =9.0144e+30, bits=  -102.83
 steps = 3, r = 6.579099405186112e-17, rel err =6.0096e+30, bits=  -102.25
 steps = 4, r = 4.386066270124075e-17, rel err =4.0064e+30, bits=  -101.66
 steps = 5, r = 2.924044180082717e-17, rel err =2.6709e+30, bits=  -101.08
 ...
 steps = 56, r = 3.057241303919973e-26, rel err =2.7926e+21, bits=  -71.242
 steps = 57, r = 2.038160869279982e-26, rel err =1.8617e+21, bits=  -70.657
 steps = 58, r = 1.358773912853321e-26, rel err =1.2412e+21, bits=  -70.072
 steps = 59, r = 9.058492752355475e-27, rel err =8.2744e+20, bits=  -69.487
 ...
 steps = 112, r = 4.209390448182399e-36, rel err =3.845e+11, bits=  -38.484
 steps = 113, r = 2.806260298791915e-36, rel err =2.5633e+11, bits=  -37.899
 steps = 114, r = 1.870840199198259e-36, rel err =1.7089e+11, bits=  -37.314
 steps = 115, r = 1.247226799469155e-36, rel err =1.1393e+11, bits=  -36.729
 ...
 steps = 166, r = 1.31498857244114e-45, rel err =   119.12, bits=  -6.8962
 steps = 167, r = 8.803082630449393e-46, rel err =   79.411, bits=  -6.3113
 steps = 168, r = 5.905213901141387e-46, rel err =    52.94, bits=  -5.7263
 steps = 169, r = 3.97330141493605e-46, rel err =   35.294, bits=  -5.1413
 ...
 steps = 221, r = 1.094764452185972e-47, rel err =2.4601e-08, bits=   25.277
 steps = 222, r = 1.094764443208569e-47, rel err =1.6401e-08, bits=   25.862
 steps = 223, r = 1.094764437223634e-47, rel err =1.0934e-08, bits=   26.447
 steps = 224, r = 1.094764433233677e-47, rel err =7.2892e-09, bits=   27.032
...
 steps = 250, r = 1.094764425253974e-47, rel err =1.9251e-13, bits=    42.24
 steps = 251, r = 1.094764425253904e-47, rel err =1.2834e-13, bits=   42.825
 steps = 252, r = 1.094764425253857e-47, rel err =8.5487e-14, bits=   43.411
 steps = 253, r = 1.094764425253826e-47, rel err =5.7065e-14, bits=   43.994
 ...
 steps = 261, r = 1.094764425253766e-47, rel err =2.2204e-15, bits=   48.678
 steps = 262, r = 1.094764425253765e-47, rel err =1.5543e-15, bits=   49.193
 steps = 263, r = 1.094764425253765e-47, rel err =1.1102e-15, bits=   49.678
 steps = 264, r = 1.094764425253764e-47, rel err =6.6613e-16, bits=   50.415
 steps = 265, r = 1.094764425253764e-47, rel err =4.4409e-16, bits=       51
 steps = 266, r = 1.094764425253764e-47, rel err =2.2204e-16, bits=       52
 steps = 267, r = 1.094764425253764e-47, rel err =2.2204e-16, bits=       52
 steps = 268, r = 1.094764425253764e-47, rel err =2.2204e-16, bits=       52
 steps = 269, r = 1.094764425253764e-47, rel err =2.2204e-16, bits=       52
 steps = 270, r = 1.094764425253764e-47, rel err =2.2204e-16, bits=       52
```

On average, the number of correct bits increases by something less than 1 at each iteration, a demonstration of linear convergence. The convergence is not quadratic due to the multiple root causing $f' = 0$ at the root. (Matlab's fsolve could not find the root accurately enough

so we inserted the exact value.)

(d) A starting value which leads to divergence is $x_0 = -2$.

A starting value which leads to convergence is $x_0 = -0.2$.

Due to the symmetry in the function, we can find an initial point x_0 such that applying one step of Newton's method causes the next value to flip about $x = \epsilon^2$:

```
function [ y, yp ] = p6d( x, p )
 y = atan( x - eps**2 );
 yp = 1 / ( 1 + ( x - eps**2 )**2;
end

convergence at x0 = -0.2:

octave:4> newton_results(-0.2, @p6d, 1, 10)
 steps = 1, r = 0.005291382243876014, rel err =1.0732e+29, bits=  -96.438
 steps = 2, r = -9.876742151481571e-08, rel err =2.0032e+24, bits=  -80.729
 steps = 3, r = 6.485096002415737e-22, rel err =1.3153e+10, bits=  -33.615
 steps = 4, r = 4.930380657631324e-32, rel err =        0, bits=      Inf
 steps = 5, r = 4.930380657631324e-32, rel err =        0, bits=      Inf
 steps = 6, r = 4.930380657631324e-32, rel err =        0, bits=      Inf
 steps = 7, r = 4.930380657631324e-32, rel err =        0, bits=      Inf
 steps = 8, r = 4.930380657631324e-32, rel err =        0, bits=      Inf
 steps = 9, r = 4.930380657631324e-32, rel err =        0, bits=      Inf
 steps = 10, r = 4.930380657631324e-32, rel err =        0, bits=      Inf

divergence at x0 = -2:
octave:5> newton_results(-2, @p6d, 1, 10)
 steps = 1, r =    3.535743588970452, rel err =7.1713e+31, bits=  -105.82
 steps = 2, r =   -13.95095908692749, rel err =2.8296e+32, bits=   -107.8
 steps = 3, r =    279.3440665336173, rel err =5.6658e+33, bits=  -112.13
 steps = 4, r =   -122016.9989179545, rel err =2.4748e+36, bits=   -120.9
 steps = 5, r =    23386004197.93385, rel err =4.7432e+41, bits=  -138.44
 steps = 6, r = -8.590766671950354e+20, rel err =1.7424e+52, bits=  -173.54
 steps = 7, r = 1.159267669890725e+42, rel err =2.3513e+73, bits=  -243.73
 steps = 8, r = -2.110995587610979e+84, rel err =4.2816e+115, bits=  -384.12
 steps = 9, r = 6.999943395317565e+168, rel err =1.4198e+200, bits=  -664.89

unstable oscillation near 1.391745242913428:
octave:2> format long e
octave:3> fsolve( @(x) 2*x - 2*eps**2 - ( 1 + ( x - eps**2 )**2 ) * atan( x - eps**2 ),
ans =    1.391745242913428e+00
octave:4> newton_results(1.391745242913428, @p6d, 1, 10)
 steps = 1, r =   -1.391745312764084, rel err =2.8228e+31, bits=  -104.48
 steps = 2, r =    1.391745497033274, rel err =2.8228e+31, bits=  -104.48
```

```
steps = 3, r =   -1.391745983143807, rel err =2.8228e+31, bits=  -104.48
steps = 4, r =    1.391747265526037, rel err =2.8228e+31, bits=  -104.48
steps = 5, r =   -1.391750648513322, rel err =2.8228e+31, bits=  -104.48
steps = 6, r =    1.391759573022311, rel err =2.8228e+31, bits=  -104.48
steps = 7, r =   -1.391783116530534, rel err =2.8229e+31, bits=  -104.48
steps = 8, r =    1.391845227115933, rel err =2.823e+31, bits=  -104.48
steps = 9, r =   -1.392009089818324, rel err =2.8233e+31, bits=  -104.48
steps = 10, r =    1.392441451784854, rel err =2.8242e+31, bits=  -104.48
```