Solution 1 We are going to prove that \( g(p) = p \iff f(p) = 0 \), where we need the assumption \( p \geq 0 \) for parts a, b and c.

a. \( g(p) = p \iff p^4 = 3 + p - 2p^2 \iff f(p) = 0 \).

b. \( g(p) = p \iff 2p^2 = 3 + p - p^4 \iff f(p) = 0 \).

c. \( g(p) = p \iff p^2(p^2 + 2) = p + 3 \iff f(p) = 0 \).

d. \( g(p) = p \iff 4p^4 + 4p^2 - p = 3p^4 + 2p^2 + 3 \iff f(p) = 0 \).

Solution 2

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 \( O(2^{-n}) \) 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 \(\{(b, c) \mid -b^2 \leq c \leq \frac{3}{4}b^2\}\) is as follows:
Solution 3

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

\[
|g'(x)| = \left| \frac{c}{x^2} \right| \leq \frac{1}{2}
\]

iff \( x^2 \geq 2|c| \).

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

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

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

c. 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, \( x_n = x^* + O(2^{-n}) \).

Altogether, the region is \( \{(b, c) \mid b^2 \geq \frac{9}{2}|c| \} \). 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 $|x_\alpha||x_\beta| \geq (\sqrt{\frac{2}{3}|c|})^2 = 2|c|$, 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
\[
x_n = \frac{-b + \sqrt{b^2 - 4c}}{2} \geq \frac{3}{\sqrt{2}}\sqrt{|c|} + \sqrt{\left(\frac{-3}{\sqrt{2}}\sqrt{|c|}\right)^2 - 4|c|} = \sqrt{2|c|}.
\]

**Solution 4** The line tangent to the graph of \( f \) at a point \( p \) is given by \( L_p(x) = f(p) + f'(p)(x-p) \). The \( x \) intercept of \( L_{p_{n-1}} \) is given by its root, i.e. \( -f(p_{n-1}) = f'(p_{n-1})(p_n-p_{n-1}) \), or
\[
p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}.
\]

**Solution 5**

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 \( x_0 \in (0, \frac{2}{a}) \) 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

\[ \frac{|x_{n+1} - \frac{1}{a}|}{|x_n - \frac{1}{a}|^2} = a. \]

Solution 6 We implemented this iterative method in MATLAB as `cubic_method.m`:

```matlab
function cubic_method(x0, f, df, ddf, n)
    % x0: initial estimate of the root
    % f: function handle
    % df: function handle for derivative of f
    % ddf: function handle for the second derivative of f
    % n: number of steps

    r = x0;
    history = [0, x0];

    for k = 1:n
        newton_step = f(r)./df(r);
        correction = 0.5 * (ddf(r)./df(r)) * (f(r)./df(r)).^2;
        r = r - newton_step - correction;
        history = [history; [k, r]];
    end

    disp(table(history(:,1), history(:,2), 'VariableNames', ...
        {'step', 'p_n'}))
end
```

The output was as follows:

```plaintext
>> cubic_method(pi/4, @(x) cos(x) - x, @(x) -sin(x) - 1, @(x) -cos(x), 5)
step  p_n
     --  ------------------
     0  0.785398163397448
```
As in Example 1 of Section 2.3 of the textbook, because of the agreement of $p_2$ and $p_3$ from the new method, we could reasonably expect this result to be accurate to the places listed. Compared to this method, Newton’s method takes one more step of iteration to achieve that accuracy.

Note: Example 1 of Section 2.4 of the textbook displays linear convergence to a multiple root since $e^x - 1 - x = O(x^2)$ as $x \to 0$, yielding an uninformative comparison.

Solution 7

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

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

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

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

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

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

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

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.
Solution 8 Sample code follows:

```matlab
function r = newton(x0, f, p, n)
%NEWTON Performs Newton's method
% 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
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:

```matlab
function newton_table(x0, f, p, n)
%NEWTON_TABLE Tabulates the results of Newton's method
% 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
% Make a function which does not output the derivative
function y = f_no_deriv(x)
    [y, ~] = f(x, p);
end

matlab_root = fzero(@(x) f_no_deriv(x), x0);
newton_roots = zeros(n+1, 1);
newton_roots(1) = x0;

for k = 1:n
    newton_roots(k+1) = newton(x0, f, p, k);
end

% Relative error (interesting when matlab_root is not too close to 0)
relative_error = abs(matlab_root - newton_roots)/matlab_root;

% Find the significand, and number of correct bits
[F_true, E_true] = log2(matlab_root);
```
\[
[F_{\text{newton}}, E_{\text{newton}}] = \log_2(\text{newton\_roots});
\]
\[
sig_{\text{true}} = 2 \times F_{\text{true}};
\]
\[
sig_{\text{newton}} = 2 \times F_{\text{newton}};
\]
\[
correct\_bits = \text{floor}(-\log_2(\max(\text{abs}(sig_{\text{true}} - sig_{\text{newton}}), \text{eps})));
\]
% Anywhere the exponent or sign does not match, set correct\_bits = 0
\[
correct\_bits(E_{\text{true}} \neq E_{\text{newton}} \text{ | sign(F_{\text{true}}) \neq sign(F_{\text{newton}})}) = 0;
\]
\[
\text{disp(table((0:n)', newton\_roots, matlab\_root*ones(n+1,1), ...}
\]
relative\_error, correct\_bits, 'VariableNames', ...
\[
{['\text{step}', '\text{newton}', '\text{true}', '\text{relative\_error}', '\text{correct\_bits'}])}
\]

We find the two roots as follows:

\[
\text{>> newton\_table(.2, @(x, p) deal(1./x + log(x) - 2, -1./x.^2 + 1./x), 1, 10)}
\]

<table>
<thead>
<tr>
<th>step</th>
<th>newton</th>
<th>true</th>
<th>relative_error</th>
<th>correct_bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.2</td>
<td>0.31784</td>
<td>0.37076</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.26953</td>
<td>0.31784</td>
<td>0.15201</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0.30922</td>
<td>0.31784</td>
<td>0.027137</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>0.31756</td>
<td>0.31784</td>
<td>0.00089972</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>0.31784</td>
<td>0.31784</td>
<td>9.9778e-07</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>0.31784</td>
<td>0.31784</td>
<td>1.2274e-12</td>
<td>39</td>
</tr>
<tr>
<td>6</td>
<td>0.31784</td>
<td>0.31784</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>7</td>
<td>0.31784</td>
<td>0.31784</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>8</td>
<td>0.31784</td>
<td>0.31784</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>9</td>
<td>0.31784</td>
<td>0.31784</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>10</td>
<td>0.31784</td>
<td>0.31784</td>
<td>0</td>
<td>52</td>
</tr>
</tbody>
</table>

\[
\text{>> newton\_table(5, @(x, p) deal(1./x + log(x) - 2, -1./x.^2 + 1./x), 1, 10)}
\]

<table>
<thead>
<tr>
<th>step</th>
<th>newton</th>
<th>true</th>
<th>relative_error</th>
<th>correct_bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>6.3054</td>
<td>0.20703</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>6.191</td>
<td>6.3054</td>
<td>0.01814</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>6.3046</td>
<td>6.3054</td>
<td>0.00013404</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>6.3054</td>
<td>6.3054</td>
<td>7.2901e-09</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>6.3054</td>
<td>6.3054</td>
<td>1.4086e-16</td>
<td>52</td>
</tr>
<tr>
<td>5</td>
<td>6.3054</td>
<td>6.3054</td>
<td>1.4086e-16</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td>6.3054</td>
<td>6.3054</td>
<td>1.4086e-16</td>
<td>52</td>
</tr>
<tr>
<td>7</td>
<td>6.3054</td>
<td>6.3054</td>
<td>1.4086e-16</td>
<td>52</td>
</tr>
<tr>
<td>8</td>
<td>6.3054</td>
<td>6.3054</td>
<td>1.4086e-16</td>
<td>52</td>
</tr>
<tr>
<td>9</td>
<td>6.3054</td>
<td>6.3054</td>
<td>1.4086e-16</td>
<td>52</td>
</tr>
</tbody>
</table>
10  6.3054  6.3054  1.4086e-16  52

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

Solution 9  We find the root as follows:

```
>> newton_table(0.2, @(x, p) deal((x - 0.111).^3, 3 * (x - 0.111).^2), 1, 25)
```

<table>
<thead>
<tr>
<th>step</th>
<th>newton</th>
<th>true</th>
<th>relative error</th>
<th>correct bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.2</td>
<td>0.111</td>
<td>0.8018</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.17033</td>
<td>0.111</td>
<td>0.53453</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.15056</td>
<td>0.111</td>
<td>0.35636</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.13737</td>
<td>0.111</td>
<td>0.23757</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0.12858</td>
<td>0.111</td>
<td>0.15838</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.12272</td>
<td>0.111</td>
<td>0.10559</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>0.11881</td>
<td>0.111</td>
<td>0.070391</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>0.11621</td>
<td>0.111</td>
<td>0.046928</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>0.11447</td>
<td>0.111</td>
<td>0.031285</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>0.11332</td>
<td>0.111</td>
<td>0.020857</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>0.11254</td>
<td>0.111</td>
<td>0.013904</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>0.11203</td>
<td>0.111</td>
<td>0.0092696</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>0.11169</td>
<td>0.111</td>
<td>0.0061798</td>
<td>6</td>
</tr>
<tr>
<td>13</td>
<td>0.11146</td>
<td>0.111</td>
<td>0.0041198</td>
<td>7</td>
</tr>
<tr>
<td>14</td>
<td>0.1113</td>
<td>0.111</td>
<td>0.0027466</td>
<td>7</td>
</tr>
<tr>
<td>15</td>
<td>0.1112</td>
<td>0.111</td>
<td>0.001831</td>
<td>8</td>
</tr>
<tr>
<td>16</td>
<td>0.1114</td>
<td>0.111</td>
<td>0.0012207</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>0.11109</td>
<td>0.111</td>
<td>0.0008138</td>
<td>9</td>
</tr>
<tr>
<td>18</td>
<td>0.11106</td>
<td>0.111</td>
<td>0.00054253</td>
<td>10</td>
</tr>
<tr>
<td>19</td>
<td>0.11104</td>
<td>0.111</td>
<td>0.00036169</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>0.11103</td>
<td>0.111</td>
<td>0.00024112</td>
<td>11</td>
</tr>
<tr>
<td>21</td>
<td>0.11102</td>
<td>0.111</td>
<td>0.00016075</td>
<td>11</td>
</tr>
<tr>
<td>22</td>
<td>0.11101</td>
<td>0.111</td>
<td>0.00010717</td>
<td>12</td>
</tr>
<tr>
<td>23</td>
<td>0.11101</td>
<td>0.111</td>
<td>7.1444e-05</td>
<td>12</td>
</tr>
<tr>
<td>24</td>
<td>0.11101</td>
<td>0.111</td>
<td>4.763e-05</td>
<td>13</td>
</tr>
<tr>
<td>25</td>
<td>0.111</td>
<td>0.111</td>
<td>3.1753e-05</td>
<td>14</td>
</tr>
</tbody>
</table>

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 at $x = 0.111$, causing $f'(0.111) = 0$.

Solution 10  A starting value which leads to divergence is $x_0 = -2$:

```
>> newton_table(-2, @(x, p) deal(atan(x - 0.111), 1./(1 + (x - 0.111).^2)), 1, 5)
```

<table>
<thead>
<tr>
<th>step</th>
<th>newton</th>
<th>true</th>
<th>relative_error</th>
<th>correct_bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

11
The negative number of correct bits signifies that the approximation does not have the correct sign.

A starting value which leads to convergence is $x_0 = 0.08$:

```matlab
>> newton_table(.08, @(x, p) deal(atan(x - 0.111), 1./(1 + (x - 0.111).^2)), 1, 5)
```

<table>
<thead>
<tr>
<th>step</th>
<th>newton</th>
<th>true</th>
<th>relative_error</th>
<th>correct_bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.08</td>
<td>0.111</td>
<td>0.27928</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0.11102</td>
<td>0.111</td>
<td>0.00017889</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>0.111</td>
<td>0.111</td>
<td>4.7009e-14</td>
<td>43</td>
</tr>
<tr>
<td>3</td>
<td>0.111</td>
<td>0.111</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>4</td>
<td>0.111</td>
<td>0.111</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>5</td>
<td>0.111</td>
<td>0.111</td>
<td>0</td>
<td>52</td>
</tr>
</tbody>
</table>

We see the convergence is very fast here.

Due to the symmetry in the function, we can find a point $x_0 \approx 1.5027$ or $-1.2807$ such that applying one step of Newton’s method causes the next value to flip about $x = 0.111$:

```matlab
>> x0 = fzero(@(x) 2*x - 2*0.111 - (1 + (x - 0.111).^2).*atan(x - 0.111), 1)
newton_table(x0, @(x, p) deal(atan(x - 0.111), 1./(1 + (x - 0.111).^2)), 1, 5)
```

```
x0 =

1.5027
```

<table>
<thead>
<tr>
<th>step</th>
<th>newton</th>
<th>true</th>
<th>relative_error</th>
<th>correct_bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.5027</td>
<td>0.111</td>
<td>12.538</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>-1.2807</td>
<td>0.111</td>
<td>12.538</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1.5027</td>
<td>0.111</td>
<td>12.538</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>-1.2807</td>
<td>0.111</td>
<td>12.538</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1.5027</td>
<td>0.111</td>
<td>12.538</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>-1.2807</td>
<td>0.111</td>
<td>12.538</td>
<td>0</td>
</tr>
</tbody>
</table>