3.1

Problem 2 To help determine the roots of $x = \tan(x)$, graph $y = x$ and $y = \tan(x)$, and look at the intersection points of the two graphs.
(a) Find the smallest nonzero positive root of $x = \tan(x)$, with an accuracy of $\epsilon = 0.0001$.
Note: The root is greater than $\pi/2$.
(b) Solve $x = \tan(x)$ for the root which is closest to $x = 100$.

Solution: (a) Both $y = x$ and $y = \tan(x)$ pass through the origin. In addition, their slopes are $y' = 1$ and $y' = \sec^2 x$, respectively. Then on $0 < x < \pi/2$, $\sec^2 x > 1$, and the function $y = \tan(x)$ increases more rapidly than $y = x$. Thus two can not intersect in $0 < x < \pi/2$. From a graph, or by checking values, it is easy to see that the two graphs must intersect on $(\pi/2, 3\pi/2)$. Using $[a, b] = [4, 4.7]$, we find a root of 4.49347.

(b) The root nearest to 100 is close to $63\pi/2$, which is slightly larger than 98.96. (In general, there are roots of $x = \tan(x)$ at approximately $(n + 0.5)\pi$, for all integers n.) Using the initial interval [98, 98.96], we have a root$\doteq$ 98.95009.

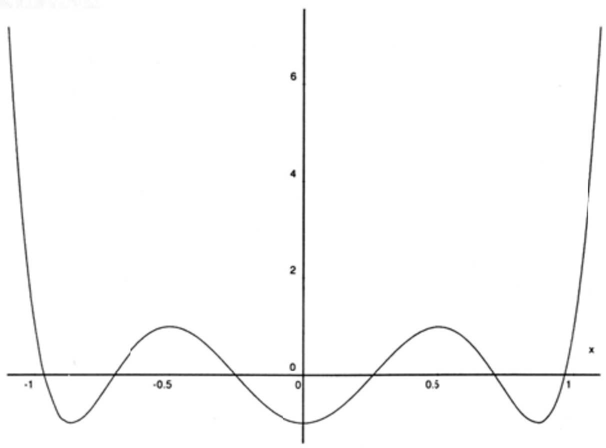
Problem 6 Using the bisection method and a graph of $f(x)$, find all roots of

$$f(x) = 32x^6 - 48x^4 + 18x^2 - 1$$

The true roots are

$$\cos\left[(2j-1)\frac{\pi}{12}\right], \qquad j = 1, 2, \ldots, 6$$

Solution: Since $f(x)$ is a polynomial of degree 6, $f(x)$ has six roots. From the following graph, for each root, the initial interval can be chosen. We compute these six roots with an accuracy of $\epsilon = 10^{-4}$.



x vs. $f(x)$ and $-1.1 \le x \le 1.1$

Initial Interval	Root
[−1, −0.75]	−0.96588
[−0.75, −0.5]	−0.70709
[−0.3, 0]	−0.25876
[0, 0.3]	0.25876
[0.5, 0.75]	0.70709
[0.75, 1]	0.96588

Problem 10 Consider the equation $e^{-x} = \sin x$. Find an interval $[a, b]$ that contains the root. Estimate the number of midpoints c needed to obtain an approximate root that is accurate within an error tolerance of 10^{-10}.

Solution: Considering the equation

$$f(x) \equiv e^{-x} - \sin x$$

Let $[a, b] = [0, 1.5]$. Using (3.9),

$$n \ge \frac{\ln\left(\frac{b-a}{\epsilon}\right)}{\ln 2} = \frac{\ln\left(\frac{1.5}{10^{-10}}\right)}{\ln 2} = 33.8$$

So we need at least $n = 34$ midpoint calculations.

ME 400/500
HW#4

Problem 15 Let $f(x) = 1 - zx$ for some $z > 0$. Solving $f(x)$ is equivalent to calculating $1/z$, thus doing a division.
(a) Give an interval $[a, b]$, or a way to calculate it, guaranteed to contain $1/z$. Do not use division in calculating or defining $[a, b]$.
(b) Assume $1 < z < 2$. Using some interval enclosing $1/z$, give the number of subdivisions n needed to obtain an estimate of $1/z$ within an accuracy of 2^{-25}.
(c) For a general $z > 0$, consider calculating $1/z$ in the single precision arithmetic of the IEEE standard floating-point arithmetic. Using the bisection method, give a way to calculate $1/z$ to the full accuracy of the arithmetic.

Solution: (a) Since $f(x) = 1 - zx$ is a monotone decreasing function and $f(0) = 1$, the following procedure can give us a interval which contains $1/z$.

step 1 If $f(1) = 0$, the root is 1.
If $f(1) < 0$ then the interval $[a, b] = [0, 1]$

step 2 Let $n_0 = 1$.

step 3 Let $n_1 = 2 * n_0$.
If $f(n_1) = 0$, the root is n_1.
If $f(n_1) < 0$ then the interval $[a, b] = [n_0, n_1]$

step 4 Let n_0 equal to n_1 and go to step 3.

This procedure is not very efficient, but it can always find the interval $[a, b]$ which contains $1/z$.

(b) Since $1 < z < 2$, $0.5 < (1/z) < 1$. Thus the interval $[0.5, 1]$ contains $1/z$. Use (3.9),

$$n \geq \frac{\ln\left[\frac{b-a}{\epsilon}\right]}{\ln 2} = \frac{\ln\left[\frac{0.5}{2^{-25}}\right]}{\ln 2} = \frac{\ln 2^{24}}{\ln 2} = 24$$

Thus we must have $n = 24$ iterates to obtain an estimate of $1/z$ within an accuracy of 2^{-25}.

(c) Assume that we are using a binary computer with IEEE standard floating-point arithmetic. Then

$$z = \hat{z} \cdot 2^e$$

where $1 \leq \hat{z} < 2$. The root can be found in following step
1. Find the interval that $f(2^k)f(2^{k+1}) < 0$.
2. Use the bisections method to subdivide the interval not more than $e - k$ times to get the solution or an approximation of the solution of the machine accuracy.

3.2

Problem 3 (a) On most computers, the computation of $\sqrt{a}$ is based on Newton's method. Set up the Newton iteration for solving $x^2 - a = 0$, and show that it can be written in the form

$$x_{n+1} = \frac{1}{2}\left[x_n + \frac{a}{x_n}\right], \qquad n \geq 0$$

(b) Derive the error and relative error formulas

$$\sqrt{a} - x_{n+1} = -\frac{1}{2x_n}(\sqrt{a} - x_n)^2$$

$$\text{Rel}(x_{n+1}) = -\frac{\sqrt{a}}{2x_n}\left[\text{Rel}(x_n)\right]^2$$

Hint: Apply (3.19)
(c) For x_0 near $\sqrt{a}$, the last formula becomes

$$\text{Rel}(x_{n+1}) \doteq -\frac{1}{2}\left[\text{Rel}(x_n)\right]^2, \qquad n \geq 0$$

Assuming $\text{Rel}(x_0) = 0.1$, use this formula to estimate the relative error in x_1, x_2, x_3, and x_4.

Solution: (a) Let $f(x) = x^2 - a$. Then $f'(x) = 2x$, and Newton's method is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - a}{2x_n} = \frac{1}{2}(x_n + \frac{a}{x_n}), \qquad n = 0, 1, \ldots$$

(b)

$$\sqrt{a} - x_{n+1} = \sqrt{a} - \frac{1}{2}(x_n + \frac{a}{x_n}) = \frac{-1}{2x_n}(a - 2\sqrt{a}\,x_n + x_n^2) = \frac{-1}{2x_n}(\sqrt{a} - x_n)^2$$

$$\text{Rel}(x_{n+1}) = \frac{\sqrt{a} - x_{n+1}}{\sqrt{a}} = \frac{-\sqrt{a}}{2x_n}\left(\frac{\sqrt{a} - x_n}{\sqrt{a}}\right)^2 = \frac{-\sqrt{a}}{2x_n}[\text{Rel}(x_n)]^2$$

(c)

$$\begin{aligned}\text{Rel}(x_1) &\doteq -5.00 \times 10^{-3}\\ \text{Rel}(x_2) &\doteq -1.25 \times 10^{-5}\\ \text{Rel}(x_3) &\doteq -7.81 \times 10^{-11}\\ \text{Rel}(x_4) &\doteq -3.05 \times 10^{-21}\end{aligned}$$

Problem 5 (a) Repeat Problem 2 of Section 3.1, for finding roots of $x = \tan x$. Use an error tolerance of $\epsilon = 10^{-6}$.
(b) The root near 100 will be difficult to find using Newton's method. To explain this, compute the quantity M of (3.21), and use it in the condition (3.24) for x_0.

Solution: See the earlier discussion in Problem 2 of Section 3.1. For convergence with Newton's method, x_0 must be chosen fairly accurately in both cases, but especially case (b).

(a) Newton's method will converge if x_0 is in the interval $4.288 \leq \alpha < (3\pi/2) \doteq 4.712$. With an error tolerance of $\epsilon = 10^{-6}$ and $x_0 = 4.288$, Newton's method converged in 14 iterations to $\alpha \doteq 4.4934096$.

(b) Newton's method will converge if x_0 is in the interval $98.94 \leq x_0 < (63\pi/2) \doteq 98.9601$. With an error tolerance of $\epsilon = 10^{-6}$ and $x_0 = 98.9401$, Newton's method converged in 10 iterations to $\alpha \doteq 98.9500656$. The true root is 98.950062824.

In order to make Newton's method convergent, we must have

$$| M(\alpha - x_0) | < 1$$

where M is defined in (3.21). Compute M,

$$M = \frac{-f''(\alpha)}{-2f'(\alpha)} = \frac{-\sec^2\alpha}{\tan\alpha} \doteq -98.96012648$$

which explains the difficulty of finding root near 100 by using Newton's method.

Problem 6 The equation

$$f(x) \equiv x + e^{-Bx^2}\cos x = 0, \qquad B > 0$$

has a unique root, and it is in the interval $(-1, 0)$. Use Newton's method to find it as accurately as possible. Use value of $B = 1, 5, 10, 25, 50$. Among your choices of x_0, choose x_0, and explain the behavior observed in the iterates for the larger values of B.
Hint: Draw a graph of $f(x)$ to better understand the behavior of the function.

Solution: As the constant B increases, the root moves towards 0. At $x = 0$, $f(0) = 1$, $f'(0) = 1$. Thus if $x_0 = 0$, then always $x_1 = -1.0$. For large values of B, $f(-1) \doteq -1.0$, $f'(-1) \doteq 1.0$; and thus, the iterate $x_1 = -1.0$ will lead to $x_2 = 0$. Thus we have an infinite loop for large values of B, if $x_0 = 0$.

In the following table, we give the results of subroutine NEWTON with $\epsilon = 10^{-8}$ and a limit on the number of iterates of 30. When the error test was not satisfied, the final iterate is given.

B	x_0	Root or x_{30}	ITERATE
1	0.0	−0.588401794	5
1	−0.5	−0.588401794	3
25	0.0	−1.000000000	30
25	−0.5	−0.237436250	30
25	−0.2	−0.237436250	30

ME 400/500
HW#4

Problem 12 Recall the material of Section 1.3 on nested evaluation of polynomials. In particular, recall (1.35)–(1.38) and Problem 8. Using this, write a Newton program for finding the roots of polynomials $p(x)$, using this earlier material to efficiently evaluate p and p'. Apply it to the following polynomial equations, finding their largest positive root.

(b) $32x^6 - 48x^4 + 18x^2 - 1 = 0$.
Note that the polynomial $q(x)$ satisfies $p(x) = (x - a)q(x)$ at the root α. Thus $q(x)$ can be used to obtain the remaining roots of $p(x) = 0$. This is called ***polynomial deflation.***

(b) The largest positive root is about 1. The initial guess is 1.0 with an error tolerance of $\epsilon = 10^{-6}$. The root is 0.9659258 with 4 iterates. The deflation polynomial is

$$32x^5 + 30.9096x^4 - 18.1436x^3 - 17.5254x^2 + 1.07180x + 1.03528$$

Problem 13 Consider applying Newton's method to find the root $\alpha = 0$ of $\sin x = 0$. Find an interval $[-r, r]$ for which the Newton iterates will converge to α, for any choice of x_0 in $[-r, r]$. Make r as large as possible.
Hint: Draw a graph of $y = \sin x$ and graphically interpret the placement of x_0 and x_1.

Solution: We like to find an interval $(-r, r)$ such that for any $x_0 \in (-r, r)$, $x_n \in (-r, r)\ \forall n \geq 1$. Solve the equation

$$\left| x - \frac{\sin x}{\cos x} \right| \leq |\, x \,|$$

Suppose $r > 0$, and from the graph of the function $\sin x$, we solve the equation

$$-(r - \frac{\sin r}{\cos r}) \leq r$$

$r = 1.165561185$. Thus the Newton iterates will converge to α, for any $x_0 \in [-r, r]$.

3.3

Problem 1 Using the secant method, find the roots of the equations in Problem 1 of Section 3.1. Use an error tolerance of $\epsilon = 10^{-6}$.

Solution: In all cases, $[x_0, x_1] = [0, 2]$. The error was estimated by $\alpha - x_n \approx x_{n+1} - x_n$.

(a) root = 1.8392868, number of computed iterates = 9

(b) root = 1.1284251, number of computed iterates = 5

(c) root = 0.4240310, number of computed iterates = 4

(d) root = 0.5671433, number of computed iterates = 6

(e) root = 0.5885327, number of computed iterates = 9