Math 128A Spring 2002 Handout # 10
Sergey Fomel February 19, 2002

Answers to Homework 3: Nonlinear Equations: Newton, Steffensen, and Others

---

1. Prove that the sequence

$$c_0 = 3\,; \quad c_{n+1} = c_n - \tan c_n\,, \quad n = 1, 2, \ldots \tag{1}$$

converges. Find the convergence limit and the order of convergence.

Solution:

(a) To find a candidate for the convergence limit, we can take the limit of the both sides of the recursion, as follows:

$$c = \lim_{n\to\infty} c_{n+1} = \lim_{n\to\infty} (c_n - \tan c_n) = c - \tan c\,.$$

Therefore,

$$\tan c = 0\,.$$

The root of $\tan x$ closest to $c_0 = 3$ is $c = \pi$.

(b) To prove that the iteration actually converges to π, we need to look at the derivative of $g(x) = x - \tan x$:

$$g'(x) = 1 - \frac{1}{\cos^2 x} = -\tan^2 x$$

In the interval $\frac{3}{4}\pi + \epsilon \le x \le \frac{5}{4}\pi - \epsilon$, (for some small $\epsilon > 0$)

$$|g'(x)| = \tan^2 x \le \tan^2\left(\frac{3}{4}\pi + \epsilon\right) < 1$$

Since the starting point $c_0 = 3$ belongs to the specified interval, the fixed-point iteration (1) converges according to the fixed-point theorem.

To show this in more detail, note that by the mean-value theorem,

$$\tan c_n = \tan c + \frac{1}{\cos^2 \xi_n}(c_n - c) = \tan\pi + \frac{1}{\cos^2 \xi_n}(c_n - \pi) = \frac{1}{\cos^2 \xi_n}(c_n - \pi)\,,$$

where ξ_n is some point between c_n and c. Therefore,

$$c - c_{n+1} = c - c_n + \tan c_n = (c - c_n)\left(1 - \frac{1}{\cos^2 \xi_n}\right) = -\tan^2 \xi_n\,(c - c_n)$$

We know that the starting point $c_0 = 3$ belongs to the interval $\frac{3}{4}\pi + \epsilon \le x \le \frac{5}{4}\pi - \epsilon$ where $\tan^2(x) \le \tan^2\left(\frac{3}{4}\pi + \epsilon\right) = M < 1$. The point ξ_0 should be in the same interval, therefore

$$|c - c_1| = \left|\tan^2 \xi_0\right|\,|c - c_0| \le M\,|c - c_0| < |c - c_0|\,,$$

and we see that c_1 is also in the specified interval. By induction, this will be true for all c_n, and

$$|c-c_{n+1}| \leq M\,|c-c_n| \leq M^2\,|c-c_{n-1}| \leq \ldots \leq M^{n+1}\,|c-c_0|\ .$$

Taking the limit,

$$\lim_{n\to\infty} |c-c_{n+1}| \leq \lim_{n\to\infty} M^{n+1}\,|c-c_0| = 0\,,$$

and the convergence is proved.

(c) The convergence order is 3 (cubic convergence). To show that, we can expand $\tan c_n$ in a Taylor series around $c=\pi$:

$$\tan c_n = \tan c + \frac{(c_n-c)}{\cos^2 c} + \frac{\tan c}{\cos^2 c}\,(c_n-c)^2 + \frac{3-2\,\cos^2\xi_n}{3\,\cos^2\xi_n}\,(c_n-c)^3$$

Since $\tan\pi=0$, the first and the third term are zero, and

$$\tan c_n = (c_n-c) + \frac{3-2\,\cos^2\xi_n}{3\,\cos^2\xi_n}\,(c_n-c)^3$$

Therefore,

$$c-c_{n+1} = c-c_n+c_n-c-\frac{3-2\,\cos^2\xi_n}{3\,\cos^2\xi_n}\,(c_n-c)^3 = -\frac{3-2\,\cos^2\xi_n}{3\,\cos^2\xi_n}\,(c_n-c)^3$$

and

$$\lim_{n\to\infty}\frac{|c-c_{n+1}|}{|c-c_n|^3} = \lim_{n\to\infty}\frac{3-2\,\cos^2\xi_n}{3\,\cos^2\xi_n} = \frac{3-2\,\cos^2\pi}{3\,\cos^2\pi} = \frac{1}{3}\,.$$

This can also be proved using the result of problem two.

2. Prove that if $g(x)\in C^m$ for some $m>1$ (continuous together with its derivatives to the order m), $g(c)=c$, $g'(c)=g''(c)=\ldots=g^{(m-1)}(c)=0$, $g^{(m)}(c)\neq 0$, and the fixed-point iteration

$$c_{n+1} = g(c_n) \tag{2}$$

converges to c, then the order of convergence is m.

Hint: Use the Taylor series of $g(x)$ around $x=c$.

Solution: The Taylor series takes the form

$$g(c_n) = g(c) + g'(c)\,(c_n-c) + \ldots + \frac{g^{(m-1)}(c)}{(m-1)!}\,(c_n-c)^{m-1} + \frac{g^{(m)}(\xi_n)}{m!}\,(c_n-c)^m$$

where only two terms can be different from zero:

$$g(c_n) = g(c) + \frac{g^{(m)}(\xi_n)}{m!}\,(c_n-c)^m\ ,$$

where ξ_n is some point between c and c_n. Taking the difference $c-c_{n+1}$, we see that

$$c-c_{n+1} = g(c)-g(c_n) = -\frac{g^{(m)}(\xi_n)}{m!}\,(c_n-c)^m\ .$$

Therefore,

$$\lim_{n\to\infty}\frac{|c-c_{n+1}|}{|c-c_n|^m} = \lim_{n\to\infty}\frac{\left|g^{(m)}(\xi_n)\right|}{m!} = \frac{\left|g^{(m)}(c)\right|}{m!} \neq 0\,.$$

By definition, this means that the order of convergence is m.

3. Determine the order of convergence for the following methods:

   (a) The *modified* Newton's method

$$c_{n+1} = c_n - m\,\frac{f(c_n)}{f'(c_n)} \tag{3}$$

   under the conditions $f(x) \in C^{m+1}$ $(m \geq 1)$, $f(c) = f'(c) = f''(c) = \ldots = f^{(m-1)}(c) = 0$, and $f^{(m)}(c) \neq 0$.

   Solution: The order of convergence is at least 2. The Taylor series of $f(c_n)$ around c is

$$f(c_n) = f(c) + f'(c)\,(c_n - c) + \ldots + \frac{f^{(m)}(c)}{m!}\,(c_n - c)^m + \frac{f^{(m+1)}(\xi_n)}{(m+1)!}\,(c_n - c)^{m+1}\ ,$$

   where ξ_n is a point between c and c_n, and only two terms can be different from zero:

$$f(c_n) = \frac{f^{(m)}(c)}{m!}\,(c_n - c)^m + \frac{f^{(m+1)}(\xi_n)}{(m+1)!}\,(c_n - c)^{m+1}$$

   Analogously, from the Taylor series for $f'(c_n)$, we obtain

$$f'(c_n) = \frac{f^{(m)}(c)}{(m-1)!}\,(c_n - c)^{m-1} + \frac{f^{(m+1)}(\xi_n^\star)}{m!}\,(c_n - c)^m$$

   Here ξ_n and $\xi_n^\star$ are points between c and c_n, not necessarily equal to each other. Forming the difference $c - c_{n+1}$, we get

$$c - c_{n+1} = c - c_n + m\,\frac{f(c_n)}{f'(c_n)} = \frac{f'(c_n)\,(c - c_n) + m\,f(c_n)}{f'(c_n)}$$

   Using the expressions above,

$$c - c_{n+1} = \frac{-\frac{f^{(m)}(c)}{(m-1)!}\,(c_n - c)^m - \frac{f^{(m+1)}(\xi_n^\star)}{m!}\,(c_n - c)^{m+1} + m\,\frac{f^{(m)}(c)}{m!}\,(c_n - c)^m + m\,\frac{f^{(m+1)}(\xi_n)}{(m+1)!}\,(c_n - c)^{m+1}}{\frac{f^{(m)}(c)}{(m-1)!}\,(c_n - c)^{m-1} + \frac{f^{(m+1)}(\xi_n^\star)}{m!}\,(c_n - c)^m}\ .$$

   Collecting terms in the numerator,

$$c - c_{n+1} = \frac{-\frac{f^{(m+1)}(\xi_n^\star)}{m!}\,(c_n - c)^2 + m\,\frac{f^{(m+1)}(\xi_n)}{(m+1)!}\,(c_n - c)^2}{\frac{f^{(m)}(c)}{(m-1)!} + \frac{f^{(m+1)}(\xi_n^\star)}{m!}\,(c_n - c)}$$

   In the limit,

$$\lim_{n\to\infty} \frac{|c - c_{n+1}|}{|c - c_n|^2} = \lim_{n\to\infty} \left| \frac{-\frac{f^{(m+1)}(\xi_n^\star)}{m!} + m\,\frac{f^{(m+1)}(\xi_n)}{(m+1)!}}{\frac{f^{(m)}(c)}{(m-1)!} + \frac{f^{(m+1)}(\xi_n^\star)}{m!}\,(c_n - c)} \right|$$

   By assumption, c_n converges to c as n approaches infinity. So do ξ_n and $\xi_n^\star$. Using the continuity of $f^{(m)}(x)$ and $f^{(m+1)}(x)$, we get

$$\lim_{n\to\infty} \frac{|c - c_{n+1}|}{|c - c_n|^2} = \frac{\frac{(m-1)!}{m!}\left(1 - \frac{m}{m+1}\right)\left|f^{(m+1)}(c)\right|}{\left|f^{(m)}(c)\right|} = \frac{1}{m\,(m+1)}\left|\frac{f^{(m+1)}(c)}{f^{(m)}(c)}\right|\ .$$

(b) Olver's method

$$c_{n+1} = c_n - \frac{f(c_n)}{f'(c_n)} - \frac{1}{2}\,\frac{f''(c_n)\,f(c_n)^2}{\left[f'(c_n)\right]^3} \tag{4}$$

under the conditions $f(x) \in C^4$, $f(c) = 0$, and $f'(c) \neq 0$.

Solution: The order of convergence is at least 3. To prove that, let us differentiate the function $g(x) = x - \frac{f(x)}{f'(x)} - \frac{1}{2}\frac{f''(x)\,f(x)^2}{\left[f'(x)\right]^3}$. For convenience, let us write it in the form

$$g(x) = x + f(x)\,h_1(x) + f(x)^2\,h_2(x)\,,$$

where $h_1(x) = -\frac{1}{f'(x)}$, and $h_2(x) = -\frac{1}{2}\frac{f''(x)}{\left[f'(x)\right]^3}$. The first three derivatives are

$$\begin{aligned}
g'(x) &= 1 + h_1(x)\,f'(x) + 2\,f(x)\,h_2(x)\,f'(x) + f(x)\,h_1'(x) + f(x)^2\,h_2'(x) \\
g''(x) &= 2\,h_2(x)\left[f'(x)\right]^2 + 2\,f'(x)\,h_1'(x) + 4\,f(x)\,f'(x)\,h_2'(x) + h_1(x)\,f''(x) + \\
&\quad 2\,f(x)\,h_2(x)\,f''(x) + f(x)\,h_1''(x) + f(x)^2\,h_2''(x) \\
g'''(x) &= 6\left[f'(x)\right]^2 h_2'(x) + 6\,h_2(x)\,f'(x)\,f''(x) + 3\,h_1'(x)\,f''(x) + 6\,f(x)\,h_2'(x)\,f''(x) + \\
&\quad 3\,f'(x)\,h_1''(x) + 6\,f(x)\,f'(x)\,h_2''(x) + h_1(x)\,f'''(x) + 2\,f(x)\,h_2(x)\,f'''(x) + \\
&\quad f(x)\,h_1'''(x) + f(x)^2\,h_2'''(x)
\end{aligned}$$

Evaluating them at the root c (such that $f(c) = 0$) produces

$$\begin{aligned}
g'(c) &= 1 + h_1(c)\,f'(c) = 0 \\
g''(c) &= 2\,h_2(c)\left[f'(c)\right]^2 + 2\,f'(c)\,h_1'(c) + h_1(c)\,f''(c) = 2\,h_2(c)\left[f'(c)\right]^2 + \frac{f''(c)}{f'(c)} = 0 \\
g'''(c) &= 6\left[f'(c)\right]^2 h_2'(c) + 6\,h_2(c)\,f'(c)\,f''(c) + 3\,h_1'(c)\,f''(c) + 3\,f'(c)\,h_1''(c) + h_1(c)\,f'''(c) \\
&= \frac{3\left[f''(c)\right]^2}{\left[f'(c)\right]^2} - \frac{f'''(c)}{f'(c)}
\end{aligned}$$

According to the general theorem from the second problem, this shows that the iteration $c_{n+1} = g(c_n)$ converges at least cubically:

$$\lim_{n\to\infty} \frac{|c - c_{n+1}|}{|c - c_n|^3} = \frac{\left|g'''(c)\right|}{3!} = \left|\frac{1}{2}\left(\frac{f''(c)}{f'(c)}\right)^2 - \frac{1}{6}\,\frac{f'''(c)}{f'(c)}\right|\,.$$

We could also obtain this result directly by using the Taylor series expansions.

(c) Steffensen's method

$$c_{n+1} = c_n - \frac{f(c_n)^2}{f\left[c_n + f(c_n)\right] - f(c_n)} \tag{5}$$

under the conditions $f(x) \in C^2$, $f(c) = 0$, and $f'(c) \neq 0$.

Solution: The convergence is at least quadratic.

To prove it, consider the Taylor series of $f\left[c_n + f(c_n)\right]$ around c_n:

$$f\left[c_n + f(c_n)\right] = f(c_n) + f'(c_n)\,f(c_n) + \frac{f''(\xi_n)}{2}\,f(c_n)^2\,,$$

where ξ_n is a point between c_n and $c_n + f(c_n)$. Therefore,

$$\frac{f(c_n)^2}{f\left[c_n + f(c_n)\right] - f(c_n)} = \frac{f(c_n)}{f'(c_n) + \frac{f''(\xi_n)}{2}\, f(c_n)}\;.$$

Forming the difference $c - c_{n+1}$, we obtain

$$c - c_{n+1} = c - c_n + \frac{f(c_n)}{f'(c_n) + \frac{f''(\xi_n)}{2}\, f(c_n)} = \frac{f(c_n) + f'(c_n)\,(c - c_n) + \frac{f''(\xi_n)}{2}\, f(c_n)\,(c - c_n)}{f'(c_n) + \frac{f''(\xi_n)}{2}\, f(c_n)}\;.$$

The Taylor series of $f(c)$ around c_n gives us

$$0 = f(c) = f(c_n) + f'(c_n)\,(c - c_n) + \frac{f''(\xi_n^\star)}{2}\,(c - c_n)^2\;,$$

where $\xi_n^\star$ is a point between c_n and c, not necessarily equal to ξ_n. Additionally,

$$0 = f(c) = f(c_n) + f'(\xi_n^\dagger)(c - c_n)\,,$$

where $\xi_n^\dagger$ is another point between c_n and c, not necessarily equal to ξ_n or $\xi_n^\star$. Putting it all together,

$$c - c_{n+1} = -\frac{\frac{f''(\xi_n^\star)}{2}\,(c - c_n)^2 + \frac{f''(\xi_n)}{2}\, f'(\xi_n^\dagger)\,(c - c_n)^2}{f'(c_n) + \frac{f''(\xi_n)}{2}\, f(c_n)}\;.$$

In the limit of n approaching infinity, we utilize the continuity of $f'(x)$ and $f''(x)$ to get

$$\lim_{n\to\infty} \frac{|c - c_{n+1}|}{|c - c_n|^2} = \lim_{n\to\infty} \left|\frac{\frac{f''(\xi_n^\star)}{2} + \frac{f''(\xi_n)}{2}\, f'(\xi_n^\dagger)}{f'(c_n) + \frac{f''(\xi_n)}{2}\, f(c_n)}\right| = \frac{1}{2}\left|\frac{f''(c)}{f'(c)}\right|\left|1 + f'(c)\right|\;.$$

4. (Programming) In this assignment, you will study the convergence of different methods experimentally using graphical tools. Note that the convergence limit

$$\lim_{n\to\infty} \frac{|c - c_{n+1}|}{|c - c_n|^p} = z \tag{6}$$

corresponds to the linear function

$$y = \log z + p\,x \tag{7}$$

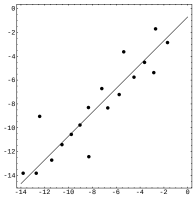
in logarithmic coordinates $x_n = \log|c - c_n|$, $y_n = |c - c_{n+1}|$. Plotting the points $\{x_n, y_n\}$ against the theoretical line verifies experimentally the order of convergence.

In the previous homework, we found that the equation

$$x + e^x = 0 \tag{8}$$

has the root at $c \approx -0.567143$ (accurate to six significant digits).

The figure shows the logarithmic plot of bisection iterations $\{x_n, y_n\}$ plotted against the line $y = \log(1/2) + x$. We can see that the iterations oscillate chaotically around the line. You will investigate whether the convergence behavior of other methods is more predictable.



Implement and apply the following methods:

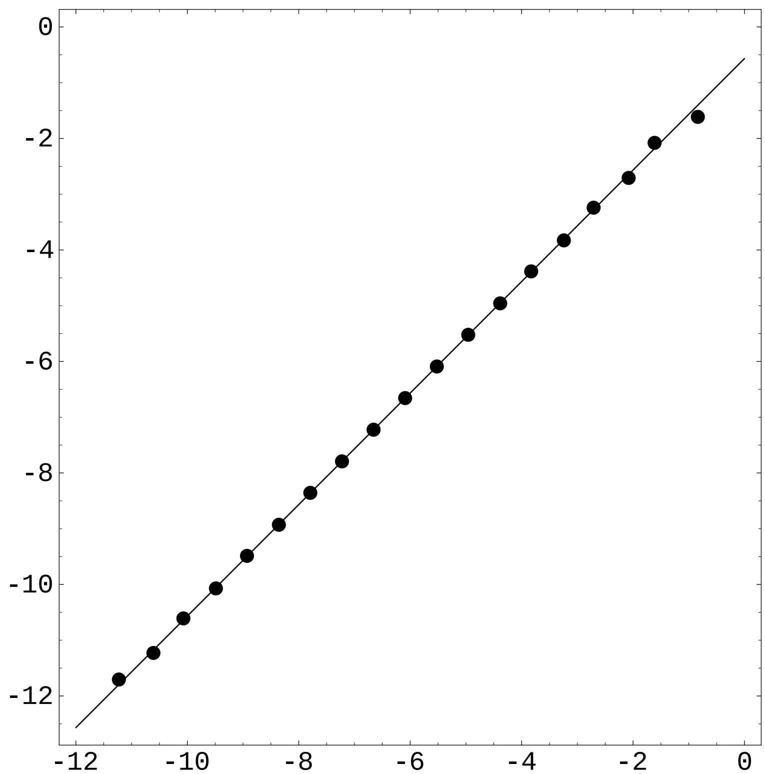
(a) Fixed-point iteration. Apply it to $g(x) = -e^x$ starting with $c_0 = -1$.

(b) Newton's method. Apply it to $f(x) = x + e^x$ starting with $c_0 = -1$.

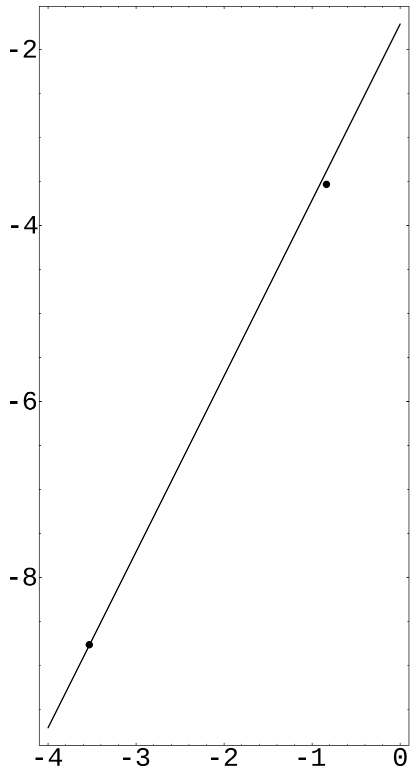
(c) Secant method. Apply it to $f(x) = x + e^x$ starting with $c_0 = 0$ and $c_1 = -1$.

In each case, find the root with the accuracy of six significant digits and plot the points x_n, y_n and the theoretical convergence line. Since some methods converge faster than others, you will need to use different number of points. Use at least 19 points for (a), 2 points for (b), and 3 points for (c).

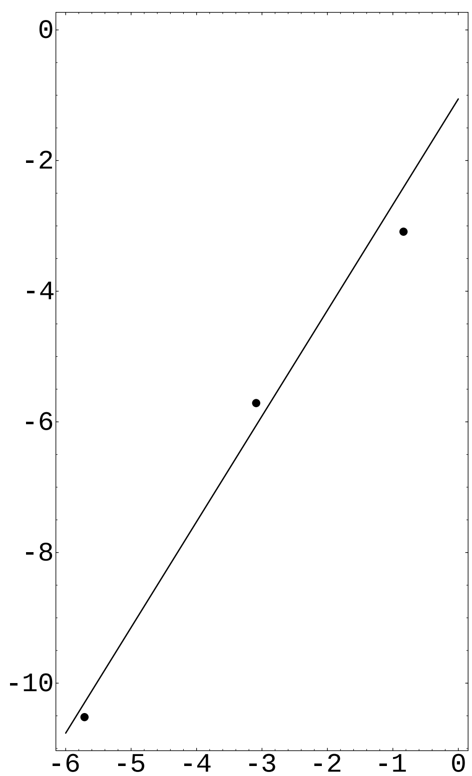
Answer:



(a)



(b)



(c)

Solution: C program

```
#include <stdio.h>  /* for output */
#include <math.h>   /* for mathematical functions */
#include <assert.h> /* for assertion */

/* function: fixed
```

```
   ---------------
   Implements fixed-point iteration
   func       - a pointer to function g(x)
   c0         - initial value
   tol        - tolerance in position and value
   nmax       - maximum number of iterations
*/
double fixed(double (*func)(double),
     double c0, double tol, int nmax)
{
  int n;
  double g, c;

  c = c0;
  for (n=0; n < nmax; n++) {
    g = func(c);

    /* print out the table */
    printf("n=%d c=%f |f(c)|=%e\n",n,c,fabs(g));

    /* return if the root is located to the tolerance */
    if (fabs(g-c) <= tol) return g;

    c = g;
  }

  fprintf(stderr,"Warning: Exact root is not found after %d iterations\n",
  nmax);
  return g;
}


/* function: newton
   ----------------
   Implements Newton’s method
   func       - a pointer to a function
   der        - a pointer to the function derivative
   c0         - initial value
   xtol, ftol - tolerance in position and value
   nmax       - maximum number of iterations
*/
double newton(double (*func)(double),
      double (*der)(double),
      double c0,
      double xtol, double ftol, int nmax)
{
  int n;
  double f, fp, c, d;

  c = c0;
  for (n=0; n < nmax; n++) {
    f = func(c);
    fp = der(c);

    assert(fp != 0.); /* avoid division by zero */
    d = f/fp;

    /* print out the table */
    printf("n=%d c=%f |f(c)|=%e\n",n,c,fabs(f));
```

```
    /* return if the root is located to the tolerance */
    if (fabs(d) <= xtol && fabs(f) <= ftol) return c;

    c -= d;
  }

  fprintf(stderr,"Warning: Exact root is not found after %d iterations\n",
  nmax);
  return c;
}


/* function: secant
   ----------------
   Implements Secant method
   func       - a pointer to a function
   c0, c1     - initial values
   xtol, ftol - tolerance in position and value
   nmax       - maximum number of iterations
*/
double secant(double (*func)(double),
      double c0, double c1,
      double xtol, double ftol, int nmax)
{
  int n;
  double f0, f1, c;

  f0 = func(c0);

  for (n=0; n < nmax; n++) {
    f1 = func(c1);

    /* print out the table */
    printf("n=%d c=%f |f(c)|=%e\n",n,c1,fabs(f1));

    /* return if the root is located to the tolerance */
    if (fabs(c1-c0) <= xtol && fabs(f1) <= ftol) return c1;

    if (c0 == c1 || f0 == f1) {
      fprintf(stderr,"Error: The line is degenerate\n");
      return c1;
    }

    c = c1 - f1*(c1-c0)/(f1-f0);

    c0 = c1;
    c1 = c;

    f0 = f1;
    f1 = func(c);

  }

  fprintf(stderr,"Warning: Exact root is not found after %d iterations\n",
  nmax);
  return c1;
}
```

```
/* test function */
static double function (double x)
{
  return (x + exp(x));
}

/* test function */
static double gfunction (double x)
{
  return (-exp(x));
}

/* derivative of the test function */
static double derivative (double x)
{
  return (1. + exp(x));
}

int main (void)
{
  int nmax=20; /* maximum number of iterations */
  double xtol=1.e-7, ftol=1.e-15, c0=-1., c1=0., c;

  c = fixed(&gfunction, c0, xtol, nmax);
  c = newton(&function, &derivative, c0, xtol, ftol, nmax);
  c = secant(&function, c1, c0, xtol, ftol, nmax);

  return 0;
}
```

5. (Programming) In this assignment, you will compute the motion of a planet according to Kepler's equation — one of the most famous nonlinear equations in the history of science. Kepler's equation has the form

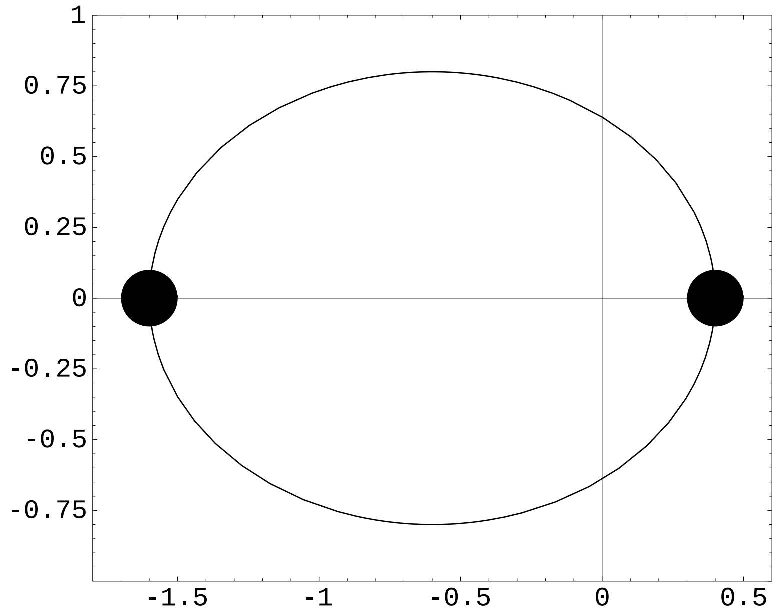
$$\omega t = \psi - \epsilon \sin\psi \; , \tag{9}$$

where t is time, ω is angular frequency, ϵ is the orbit eccentricity, and ψ is the angle coordinate. To find the planet location at time t, we need to solve equation (9) for ψ. The planet coordinates x and y are then given by

$$x = a\,(\cos\psi - \epsilon) \; ; \tag{10}$$

$$y = a\sqrt{1-\epsilon^2}\,\sin\psi \; , \tag{11}$$

where a is the major semi-axis of the elliptical orbit. For our planet, we will take $a = 1\,\mathrm{AU}$ (astronomical unit), and the eccentricity $\epsilon = 0.6$ (which is much larger than the orbit eccentricity of the Earth and other big planets in the Solar system). The picture shows the orbit and the planet positions in January ($\psi = \pi$) and July ($\psi = 0$).



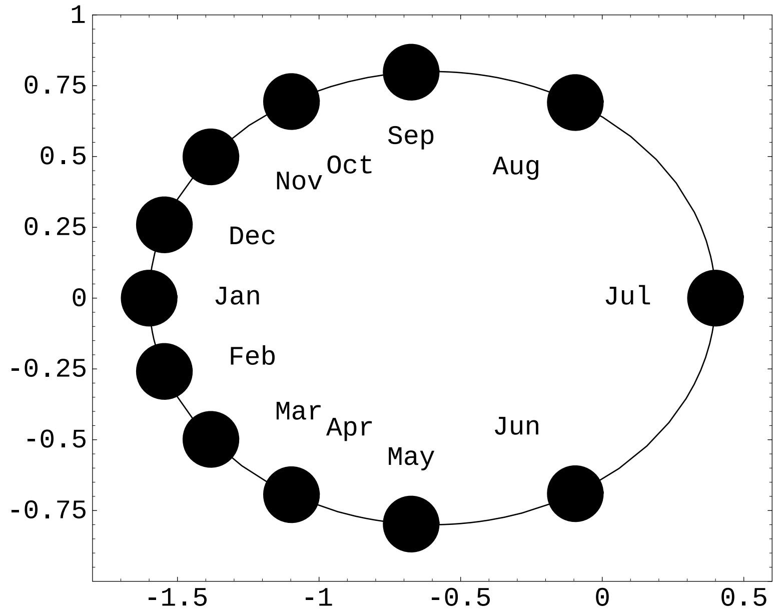
Your task is to find the planet location in the other ten months, assuming that each month takes 1/12 of the rotation period. Solve Kepler's equation (9) for $\omega t = 0, \pi/6, 2\cdot\pi/6, \ldots, 11\cdot\pi/6$. You can use any numerical method to do that (either your own program or a library program). The result should be computed with the precision of 1 second (1/3600 of 1°). Output a table of the form

ωt	ψ	x	y

and then use a graphics program to plot the planet locations.

Answer:

ωt	ψ	x	y
0.000000	0.000000	0.400000	0.000000
0.523599	1.041495	-0.095069	0.690528
1.047198	1.645523	-0.674657	0.797767
1.570796	2.091329	-1.097343	0.694043
2.094395	2.468459	-1.381872	0.498751
2.617994	2.812121	-1.546213	0.258835
3.141593	3.141593	-1.600000	0.000000
3.665191	3.471064	-1.546213	-0.258835
4.188790	3.814727	-1.381872	-0.498751
4.712389	4.191856	-1.097343	-0.694043
5.235988	4.637662	-0.674657	-0.797767
5.759587	5.241691	-0.095069	-0.690528



Solution: C program

```
#include <stdio.h> /* for output */
#include <math.h>  /* for mathematical functions */

int main (void)
{
  double x, y, t, a=1.0, c, f, fp, e=0.6, pi, tol;
  int n, iter, niter=100;

  pi = acos(-1.0);     /* number pi */
  tol = pi/180./3600.; /* accuracy */
  for (n=0; n < 13; n++) {
    t = n*2.*pi/12;
    c = t; /* the initial estimate */
    for (iter=0; iter < niter; iter++) { /* Nonlinear solver */
      /* Kepler’s equation */
      f = c - e*sin(c) - t; /* function   */
      fp = 1. - e*cos(c);   /* derivative */
      if (fabs(f) < tol && fabs(f) < tol*fabs(fp)) break;
      c -= f/fp; /* Newton’s iteration */
    }
    if (iter >= niter) {
      fprintf(stderr,
      "Newton’s method failed to converge after %d iterations\n",
      iter);
      return 1;
    }
    /* compute coordinates */
    x = a*(cos(c) - e);
    y = a*sqrt(1-e*e)*sin(c);
    /* output table */
    printf("%d t=%f psi=%f x=%f y=%f\n", n, t, c, x, y);
  }

  return 0;
}
```