

CSC112 Spring 2011
Fundamentals of Computer Science
Lab 9

Test #2 will take much of our focus of attention this week. In the interest of balancing our workload, we will do something very simple for this week's lab.

This week's lab introduces you to numerical computing, and gives you some experience with recursion. We will write a computer program to solve the non-linear equation:

$$f(x) = x^2 - \cos(7x) - \frac{3}{2} = 0 \tag{1}$$

Figure 1 shows a plot of $f(x)$.

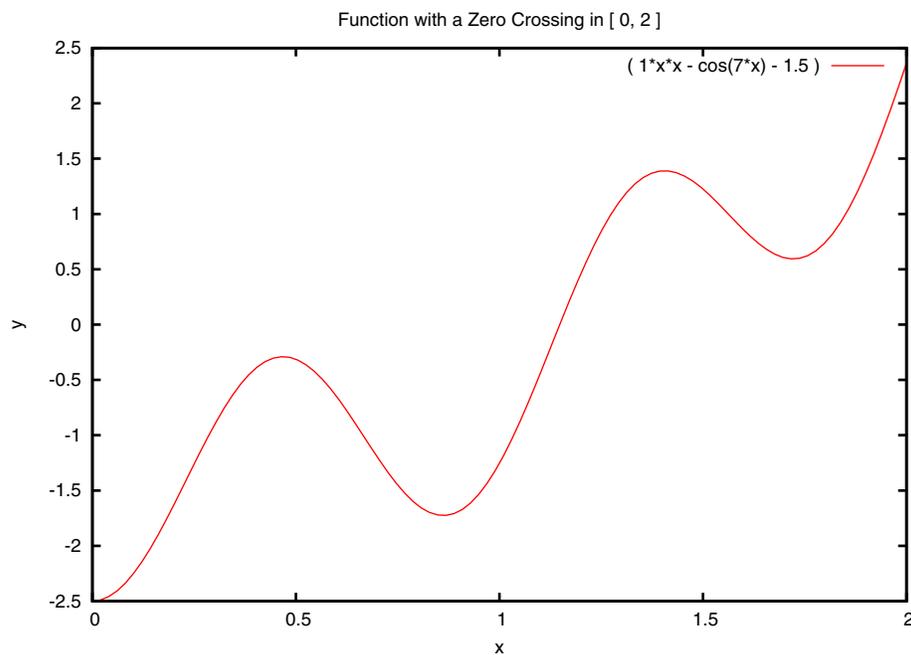


Figure 1: $f(x) = x^2 - \cos(7x) - \frac{3}{2}$

We wish to find a number x such that $f(x) = 0$. The point x is sometimes called a *zero crossing*¹. A simple method for finding a zero crossing of a continuous function is called “bisection”.

The method starts by knowing an interval $[a, b]$ in which the function changes sign. One way to guarantee a zero crossing is to check if $f(a)f(b) < 0$. In our example (shown in figure 1), $f(x)$ goes from negative to positive in the interval $[0, 2]$.

¹Steely Dan has a song with a lyric reference to a *zero crossing*.

The method is based on the following steps:

1. Compute the midpoint $u = (a + b)/2$
2. Check for the following 3 cases:
 - (a) If $f(a)f(u) < 0$ then the function changes sign in the interval $[a, u]$.
 - (b) If $f(u) == 0$ then we have found our answer.
 - (c) If $f(a)f(u) > 0$ then the function changes sign in the interval $[u, b]$.

Each time these steps are repeated, the size of the interval containing the desired answer decreases by a factor of two. After k steps, the remaining error is:

$$\varepsilon = \frac{b_0 - a_0}{2^k}$$

where $[a_0, b_0]$ is the initial interval containing the zero crossing. In a computer implementation, it is natural to update variables `a` and `b` to reflect each newly found (smaller) interval. In that case you need only check if $(b - a) < \tau$, where τ is the error tolerance.

Write a program to find an accurate approximation to the value of x where the zero-crossing occurs. Your program must include a function for $f(x)$. Your program must include a recursive function to implement the bisection method. The function header for the bisection method should be:

```
double bisection( double a, double b )
```

The base case for a recursive implementation is when $(b - a) < \tau$. Find the solution to equation (1) with an error less than $\tau = 10^{-6}$. Recall that numbers in scientific notation such as 1.0×10^{-6} can be written in C++ as `1.0e-6`. Use data type `double` throughout your computation.

Create a directory named `Lab9`. Keep all of your source and compiled programs in the directory `Lab9`.

Turn in: Change to the directory containing the sub-directory “`Lab9`”. Create a file named “`lab9.tar`” using the command:

```
tar cf lab9.tar Lab9
```

Upload the file “`lab9.tar`” to your account on telesto.