M348-53610: Scientific Computing
Homework # 04
Handout: 02/09/2016, Tuesday
Due: 02/16/2016, Tuesday

Submission. Please make your homework neat and stapled. You have to submit your homework in ECJ 1.204 before **3:00 PM** on the due date. Note that *no late homework will be accepted without compelling reasons.*

---

1 To be Graded

Problem 1. Let $f(x) = x^2 - 6$. Perform 2 steps of the Secant method to find an approximation, c_3, to the root of f, starting with the initial guesses $c_0 = 3$ and $c_1 = 2$. Include the results in all iterations, that is, c_2 and c_3, in your submission.

Problem 2. Let $f(x) = -x^3 - \cos x$. Perform 2 steps of the Secant method to find an approximation, c_3, to the root of f, starting with the initial guesses $c_0 = -1$ and $c_1 = 0$. Include the results in all iterations, that is, c_2 and c_3, in your submission.

Problem 3 (Programming Assignment)**.** A full description of the Newton's method is provided in Algorithm 2.3 of Burden and Faires (also attached here in case you do not have the textbook).
(a) Implement the Newton's method in Matlab, C++ or any other program language that you prefer. Modify the algorithm such that your program output the iteration number and the approximate root at the current iteration for each iteration of the algorithm. Include the full program in your submission.
(b) Use the program you implemented to determine a solution accurate to within $\varepsilon = 10^{-6}$ for $e^x + 2^{-x} + 2\cos x - 6 = 0$, for $x \in [1, 2]$. Include output of all iterations in your submission.

Problem 4 (Programming Assignment)**.** A full description of the Secant method is provided in Algorithm 2.4 of Burden and Faires (also attached here in case you do not have the textbook).
(a) Implement the Secant method in Matlab, C++ or any other program language that you prefer. Modify the algorithm such that your program output the iteration number and the approximate root at the current iteration for each iteration of the algorithm. Include the full program in your submission.
(b) Use the program you implemented to determine a solution accurate to within $\varepsilon = 10^{-7}$ for $x^2 - 4x + 4 - \ln x = 0$, for $x \in [2, 4]$. Include output of all iterations in your submission.

Problem 5. Show that the following sequences converge linearly to $p = 0$. How large must n be before $|p_n - p| < 5 \times 10^{-2}$?

$$(a)\ p_n = \frac{1}{n}, \quad n \geq 1 \qquad\qquad (b)\ p_n = \frac{1}{n^2}, \quad n \geq 1$$

Problem 6. Show that the Bisection algorithm gives a sequence with an error bound that converges linearly to 0.

2 Reading Assignments

- Review Sections 2.3 and 2.4 of Burden & Faires or 3.2 and 3.5 of Epperson.

Fixed-Point Iteration

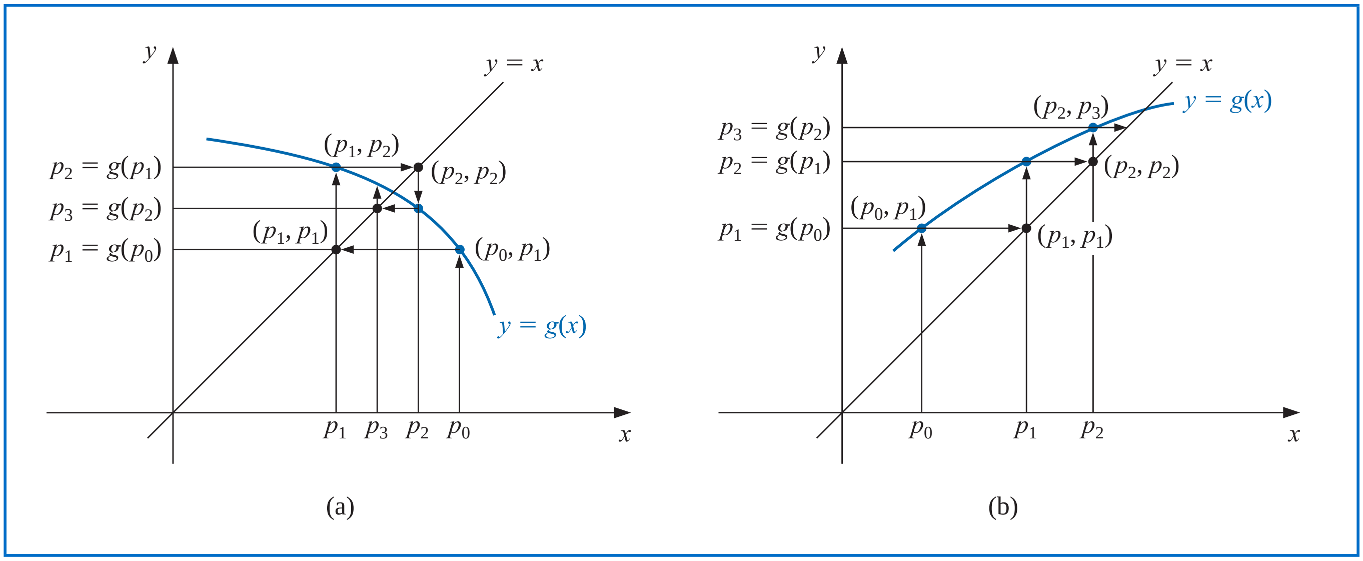
We cannot explicitly determine the fixed point in Example 3 because we have no way to solve for p in the equation $p = g(p) = 3^{-p}$. We can, however, determine approximations to this fixed point to any specified degree of accuracy. We will now consider how this can be done.

To approximate the fixed point of a function g, we choose an initial approximation p_0 and generate the sequence $\{p_n\}_{n=0}^{\infty}$ by letting $p_n = g(p_{n-1})$, for each $n \geq 1$. If the sequence converges to p and g is continuous, then

$$p = \lim_{n\to\infty} p_n = \lim_{n\to\infty} g(p_{n-1}) = g\left(\lim_{n\to\infty} p_{n-1}\right) = g(p),$$

and a solution to $x = g(x)$ is obtained. This technique is called **fixed-point**, or **functional iteration**. The procedure is illustrated in Figure 2.7 and detailed in Algorithm 2.2.

Figure 2.7



(a)

(b)

ALGORITHM 2.2

Fixed-Point Iteration

To find a solution to $p = g(p)$ given an initial approximation p_0:

INPUT initial approximation p_0; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

```
Step 1  Set $i = 1$.
Step 2  While $i \leq N_0$ do Steps 3–6.
    Step 3  Set $p = g(p_0)$.   (Compute $p_i$.)
    Step 4  If $|p - p_0| < TOL$ then
                OUTPUT ($p$);   (The procedure was successful.)
                STOP.
    Step 5  Set $i = i + 1$.
    Step 6  Set $p_0 = p$.   (Update $p_0$.)
Step 7  OUTPUT ('The method failed after $N_0$ iterations, $N_0$ =', $N_0$);
        (The procedure was unsuccessful.)
        STOP.
```
■

The following illustrates some features of functional iteration.

Illustration The equation $x^3+4x^2-10=0$ has a unique root in $[1,2]$. There are many ways to change the equation to the fixed-point form $x=g(x)$ using simple algebraic manipulation. For example, to obtain the function g described in part (c), we can manipulate the equation $x^3+4x^2-10=0$ as follows:

$$4x^2=10-x^3, \quad \text{so} \quad x^2=\frac{1}{4}(10-x^3), \quad \text{and} \quad x=\pm\frac{1}{2}(10-x^3)^{1/2}.$$

To obtain a positive solution, $g_3(x)$ is chosen. It is not important for you to derive the functions shown here, but you should verify that the fixed point of each is actually a solution to the original equation, $x^3+4x^2-10=0$.

(a) $x=g_1(x)=x-x^3-4x^2+10$

(b) $x=g_2(x)=\left(\frac{10}{x}-4x\right)^{1/2}$

(c) $x=g_3(x)=\frac{1}{2}(10-x^3)^{1/2}$

(d) $x=g_4(x)=\left(\frac{10}{4+x}\right)^{1/2}$

(e) $x=g_5(x)=x-\frac{x^3+4x^2-10}{3x^2+8x}$

With $p_0=1.5$, Table 2.2 lists the results of the fixed-point iteration for all five choices of g.

Table 2.2

n	(a)	(b)	(c)	(d)	(e)
0	1.5	1.5	1.5	1.5	1.5
1	−0.875	0.8165	1.286953768	1.348399725	1.373333333
2	6.732	2.9969	1.402540804	1.367376372	1.365262015
3	−469.7	$(-8.65)^{1/2}$	1.345458374	1.364957015	1.365230014
4	1.03×10^8		1.375170253	1.365264748	1.365230013
5			1.360094193	1.365225594	
6			1.367846968	1.365230576	
7			1.363887004	1.365229942	
8			1.365916734	1.365230022	
9			1.364878217	1.365230012	
10			1.365410062	1.365230014	
15			1.365223680	1.365230013	
20			1.365230236		
25			1.365230006		
30			1.365230013		

The actual root is 1.365230013, as was noted in Example 1 of Section 2.1. Comparing the results to the Bisection Algorithm given in that example, it can be seen that excellent results have been obtained for choices (c), (d), and (e) (the Bisection method requires 27 iterations for this accuracy). It is interesting to note that choice (a) was divergent and that (b) became undefined because it involved the square root of a negative number. □