Instructor: Emre Mengi

Math 304 (Spring 2012)

Study Guide for Weeks 1-2

Homework 1 concerns the numerical calculation of the zeros of a function, specifically the topics listed below. Next week the scanned version of Chapter 2 of the book by Burden&Faires will be available online via the library's website.

- The bisection method (Burden&Faires 2.1)
- The fixed-point iteration (Burden&Faires 2.2)
- Existence and uniqueness of a fixed-point (Burden&Faires 2.2, Theorem 2.3)
- Convergence of the fixed-point iteration (Burden&Faires 2.2, Theorem 2.4)
- Newton's method and its convergence (Burden&Faires 2.3, Theorem 2.6 for convergence)
- Secant method (Burden&Faires 2.3, pages 71-72)

Homework 1 (due on March 2nd, Friday by 5pm)

Question 9 is a bonus question. You will receive extra credit if you solve this question. In Matlab questions attach Matlab outputs as well as the print-outs of m-files.

1. (Burden&Faires, Exercise 2.1.3) Use the bisection method to find a zero of the polynomial

$$p(x) = x^4 - 2x^3 - 4x^2 + 4x + 4$$

accurate to within 10^{-1}. Perform all calculations by hand.

2. How many iterations would you need to achieve an approximation with accuracy 10^{-4} for a zero of a continuous function $f : \mathbb{R} \to \mathbb{R}$ on $[0, 1]$ by using the bisection method.

3. (Burden&Faires, Exercise 2.1.13) An implementation of the bisection method is provided on the course webpage. You should call it as follows.

```
>> zero = bisection(a,b,fname,iter)
```

Here $[a, b]$ is an interval containing a root, `fname` is a string (an array of characters) evaluating the function at a given point and `iter` is the number of bisection iterations to be performed. For instance to apply the bisection iteration 20 times to estimate a zero of $f(x) = x - e^{-x}$ on the interval $[0, 1]$ first generate an m-file `exponent.m`

```
function f = exponent(x)

f = x - exp(-x);
return;
```

then type

```
>> zero = bisection(0,1,'exponent',20)
```

in the Matlab command line.

Using the bisection routine evaluate $\sqrt[3]{25}$ correct to within 10^{-4}.

4. (Burden&Faires, Exercise 2.1.19) A trough of length L has a cross-section in the shape of a semi-circle with radius r. When filled with water to within a distance h of the top, the volume V of water is

$$V = L\left[0.5\pi r^2 - r^2 \arcsin(h/r) - h(r^2 - h^2)^{1/2}\right].$$

Suppose $L = 10$ft, $r = 1$ft, and $V = 12.4$ ft^3. Find the depth of the water in the trough to within 0.01ft. You can use the Matlab implementation of the bisection algorithm provided on the course webpage.

5. (Burden&Faires, Exercise 2.2.1-2) Use algebraic manipulations to show that each of the following functions has a fixed point at p precisely when $f(p) = 0$, where $f(x) = x^4 + 2x^2 - x - 3$.

$$\textbf{(i)}\ g_1(x) = (3 + x - 2x^2)^{1/4}, \quad \textbf{(ii)}\ g_2(x) = \left(\frac{x+3}{x^2+2}\right)^{1/2}, \quad \textbf{(iii)}\ g_3(x) = \frac{3x^4 + 2x^2 + 3}{4x^3 + 4x - 1}$$

(a) Perform three fixed-point iterations, if possible, on each of the functions g_1, g_2, g_3 starting with $p_0 = 1$.

(b) On which function do you think the fixed-point iteration converges fastest? Try to justify your observations from part **(a)** with the fixed-point theorems discussed in class.

6. (Burden&Faires, Exercise 2.2.5) An implementation of the fixed-point iteration is provided on the course webpage. It should be called as follows.

```
>> zero = fixedpoint(p0,fname,iter)
```

Above p_0 is an initial guess for a root, `fname` is a string (an array of characters) evaluating the function (f for which a root is sought) at a given point and `iter` is the number of fixed-point iterations to be performed.

Use a fixed-point iteration method to determine a zero of $p(x) = x^3 - x - 1$ on $[1, 2]$. You may have to modify the implementation of the fixed-point iteration, because it may not converge as it is. At the moment the Matlab code applies the fixed-point iteration to $g(x) = x - f(x)$. You may have to apply the fixed-point iteration to a different function.

7. (Burden&Faires, Exercise 2.2.16) Let A be a given positive constant and $g(x) = 2x - Ax^2$.

(a) Show that if fixed-point iteration converges to a non-zero limit, then the limit is $p = 1/A$, so the inverse of a number can be found using only multiplications and subtractions.

(b) Find an interval about $1/A$ for which fixed-point iteration converges, provided p_0 is in that interval.

8. (Burden&Faires, Exercise 2.2.23) An object falling vertically through the air is subjected to viscous resistance as well as to the force of gravity. Assume that an object with mass m is dropped from a height s_0 and that the height of the object after t seconds is

$$s(t) = s_0 - \frac{mg}{k}t + \frac{m^2 g}{k^2}(1 - e^{-kt/m})$$

where $g = 32.17$ ft/s^2 and k represents the coefficient of air-resistance in lb· s/ft. Suppose $s_0 = 300$ ft, $m = 0.25$ lb, and $k = 0.1$ lb·s/ft. Find, to within 0.01s, the time it takes this object to hit the ground by using a fixed-point iteration. You could use the implementation of the fixed-point iteration, or a modified version, provided on the course webpage.

9. (*) (Burden&Faires, Exercise 2.2.22) Suppose that g is continuously differentiable on some interval (c, d) that contains the fixed point p of g. Show that if $|g'(p)| < 1$, then there exists a $\delta > 0$ such that if $|p - p_0| \leq \delta$, then the fixed-point iteration converges.

10. To compute $\sqrt{7}$ one approach is to apply Newton's method to the function $f(x) = x^2 - 7$. Find the Newton update rule for $f(x) = x^2 - 7$, *i.e.*, find the relation between x_{k+1} and x_k.

11. Consider the polynomials $f(x) = x^3 + x$ and $f(x) = x^2$.

(a) Give the update rules of Newton's method and the secant method for the functions f and g.

(b) Let $\{p_k\}$ and $\{q_k\}$ be the Newton sequences for the functions f and g, respectively. Assume that both p_k and q_k converge to the root $x_* = 0$ as $k \to \infty$.

Using the update rules from part **(a)** deduce

$$\lim_{k\to\infty} \frac{|p_{k+1}|}{|p_k|^2} = c_1, \quad \text{but} \quad \lim_{k\to\infty} \frac{|q_{k+1}|}{|q_k|} = c_2$$

for some constants $c_1 \in \mathbb{R}$ and $c_2 \in (0, 1)$. The first is an example of quadratic convergence, while the second is a linear convergence. Which sequence do you think converge faster?

(c) Generate an m-file `fun.m`

```
function [f,g] = fun(x)
% Task : Computes the function x^2 and its derivative.
f = x^2;
g = 2*x;

return
```

which computes the function $g(x) = x^2$ and the derivative $g'(x) = 2x$ and stores these values in the output arguments f and g, respectively.

(d) Generate an m-file similar to the one in **(c)** to compute the function value of $f(x) = x^3 + x$ and its derivative. The name of the m-file must be same as the function name.

(e) A Matlab implementation of Newton's method `Newton.m` for zero finding is provided on the course webpage. Type `help Newton` for information about its input and output arguments. Specifically, to compute the root of a function $f : \mathbb{R} \to \mathbb{R}$, Newton's method needs the function value $f(x)$ and the derivative $f'(x)$ at every iteration. The second input argument to `Newton.m` is the name of an m-file that takes a vector $x \in \mathbb{R}^n$ as input and returns the function value $f(x)$ and the derivative $J(x) = f'(x)$ as output. The third argument is the number of Newton iterations to be applied. For instance type

```
>> [xmin,fmin] = Newton(2,'fun',10);
```

to retrieve the root of $f(x) = x^2$ starting from the initial guess $x_0 = 2$ after generating the m-file `fun.m` in **(c)**. In this case 10 Newton iterations will be applied.

Using the Matlab routine provided compute the roots of $f(x) = x^3 + x$ and $g(x) = x^2$ with the initial guesses $x_0 = 1$ and $x_0 = 4$. Include your Matlab output. Which sequence converges faster in practice? Is what you observe in harmony with what you deduced with part **(b)**.

(f) Implement the secant method by modifying the m-file `Newton.m`. Run your implementation to find the roots of $f(x) = x^3 + x$ and $g(x) = x^2$ starting from $x_0 = 2$ and $x_1 = 2.2$ (recall that the secant method requires two initial guesses). How fast does the secant method converge as compared to Newton's method for these two functions?

12. Newton's method may not converge, when the initial guess is not sufficiently close to a solution. Consider for example the polynomial $f(x) = -x^7 + x^3 + 8x$. Show by hand that the basic Newton's method does not converge with the initial guess $x_0 = 1$.