MATH 2130 Midterm Test 1 — Solutions

Prof. A. Willms, Dept. of Mathematics and Statistics, University of Guelph

Jan. 30, 2019

10 pts. 1. Suppose we want to plot a function $y = f(x)$ using n equally spaced points over some interval $I = [a, b]$.

(a) Write a MATLAB function that will take as in input the following three items:

- a handle to a function f that will take a vector x and give back a vector of y values,
- an interval I (a vector with the two entries a and b),
- a positive integer n.

Your function will first construct a vector x of n equally spaced points from a to b (you may use builtin MATLAB commands). Then your function will evaluate $y = f(x)$ and plot y vs. x. We will call the function you write `intervalplotter` so the first line of your function will start with

```
function  intervalplotter(
```

(b) How would you call your function written in (a) to plot the function $y = x^2 + 4/x$ over the interval $I = [0.2, 5]$ with $n = 200$ points?

Solution:

(a) MATLAB code for this function is shown below. (Comments are not essential.)

```
function intervalplotter(f, I, n)
% intervalplotter - plot f over an interval I using n points
%
%  intevalplotter(f,I,n)
%  INPUT:
%    f : handle to a function that takes a vector x and returns a vector y
%    I : 2-element vector defining an interval [a,b]
%    n : number of points to plot
x = linspace(I(1),I(2),n);
y = f(x);
plot(x,y)
```

(b) To call the above function in MATLAB to plot the requested curve, we would use

```
intervalplotter(@(x) x.^2 + 4./x,[0.2,5],200)
```

Alternatively, one could define the function first and then call the plotter as

```
f = @(x) x.^2 + 4./x;
intervalplotter(f,[0.2,5],200)
```

3 pts. 2. A typical floating point number system uses a string of bits (each with value 0 or 1) of the form

$$s\, c_1\, c_2\, \cdots\, c_m\, f_1\, f_2\, \cdots\, f_p$$

to represent a number. Here s is the sign bit, the c_i are the exponent bits, and the f_i are the mantissa (or fraction) bits. What number does that string of bits represent? (You can assume it is a "normal" floating point number, that is, not one of the special cases.)

Solution:
This string of bits represents the number

$$(-1)^s (1.f_1\, f_2\, f_3\, \cdots\, f_p)_2 \times 2^{c-(2^{m-1}-1)},$$

where c is the binary unsigned integer defined by the c_i bits, that is, $c = (c_1\, c_2\, \cdots\, c_m)_2$.

4 pts. 3. Give two forms for the roots of the quadratic equation $ax^2 + bx + c = 0$. Indicate when it is computationally most appropriate to use one form or the other.

Solution:
The roots of the quadratic may be given as either

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad \text{or} \qquad x = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}.$$

In order to avoid the possibility of subtracting two nearly equal numbers, when $b > 0$ it is computationally more appropriate to use the first equation with the lower sign ($-$ sign) and use the second equation with the lower upper sign ($+$ sign). Conversely if $b < 0$ we should use the first equation with the upper sign and the second equation with the lower sign. In other words use

$$x_1 = \frac{-b - \text{sign}(b)\sqrt{b^2 - 4ac}}{2a} \qquad \text{and} \qquad x_2 = \frac{-2c}{b + \text{sign}(b)\sqrt{b^2 - 4ac}}$$

5 pts. 4. Consider a fixed point iteration problem $x_{n+1} = g(x_n)$, for x_n in the interval $[a, b]$. State the conditions on $g(x)$ that guarantee that this fixed point iteration will converge to a unique solution.

Solution:
The conditions are:

(a) $g(x)$ maps the interval $[a, b]$ into itself,

(b) $g'(x)$ is continuous on $[a, b]$, and

(c) there exists a value $N < 1$ such that $|g'(x)| \leq N$ for all $x \in [a, b]$.

10 pts. 5. Briefly describe how Newton's method works.

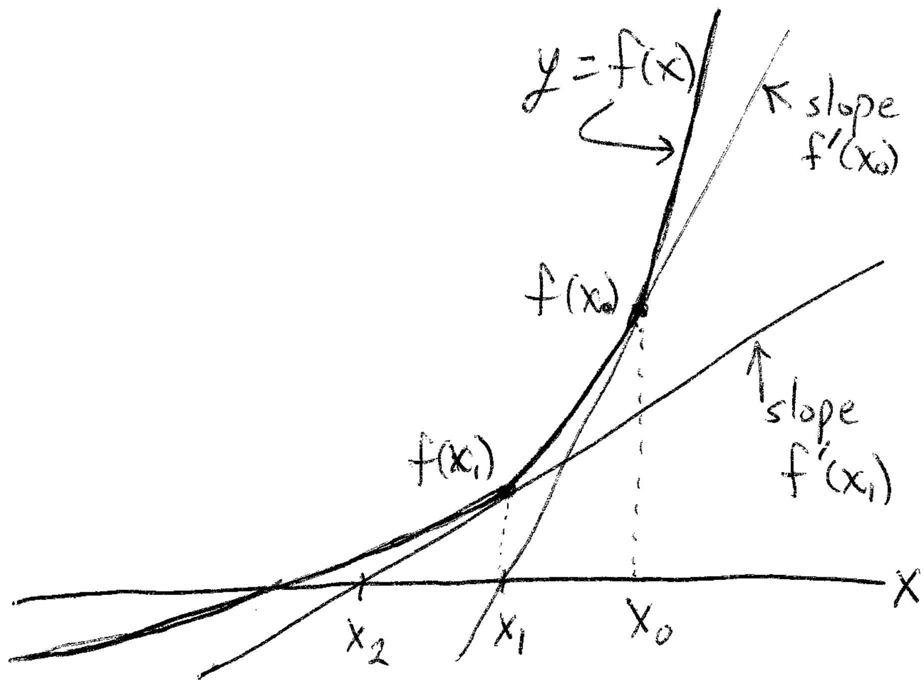
Solution:
Newton's method finds a zero of a function $f(x)$. Starting with a guess, x_0, it constructs the tangent line to the function at $(x_0, f(x_0))$ and then determines where this line crosses zero. This location is then the new guess, x_1, for the zero of $f(x)$. The method iterates in this manner until either the magnitude of the function, $|f(x_n)|$, is below some specified tolerance level, or until the specified maximum number of iterations has been reached, or until two successive guesses to the zero and within some specified relative distance between each other. The equation of the tangent line at x_n is

$$\frac{y - f(x_n)}{x - x_n} = f'(x_n),$$

hence the location x_{n+1} where this line crosses $y = 0$ is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

A picture is shown below.



10 pts. 6. The following MATLAB code is an incorrect implementation of the fixed point iteration

$$x_{n+1} = g(x_n) = \frac{1}{2}\left(x_n + \frac{5}{x_n}\right).$$

The implementation is intended to return the result (an approximation of $\sqrt{5}$) after two successive values of x are within a relative distance of `tolx`, or until a maximum number of iterations, `maxiter`, is reached.

Circle all errors and in the blank area to the right give the correct version of the lines with errors.

```
1     function fpiter(x,tolx,maxiter)
2     % fpiter - fixed point iteration for g(x)
3     iter = 0;
4     xold = x;
5     x = xold*(1 + 2*tolx);
6     while (x-xold) > tolx*abs(x) || iter < maxiter
7         iter = iter + 1;
8         x = xold;
9         x = 0.5*(xold + 0.5/xold);
10     end
```

Solution:
Errors were on lines 1, 6 (two errors), 8, and 9. Their corrected versions are below.

```
1     function x = fpiter(x,tolx,maxiter)

6     while abs(x-xold) > tolx*abs(x) && iter < maxiter

8         xold = x;
9         x = 0.5*(xold + 5.0/xold);
```

4 pts. 7. Suppose you wish to use the Golden-Section Search Algorithm to find the minimum of $f(x)$. Suppose further that the first three points are $x_1 = 1$, $x_2 = 3$, and $x_3 = 4$, and that $r = (3-\sqrt{5})/2 \approx 0.38$. (The Golden-Section Search Algorithm uses r to determine the next value x_4.)

(a) Approximately what is the value of x_4?

(b) Given the table of function values below, which three points among x_1, x_2, x_3, x_4 will be the new "bracketing triple".

x	1	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.2	2.3	2.4	2.5	2.6	2.7
$f(x)$	3.82	3.28	3.01	2.85	2.62	2.41	2.22	2.33	2.48	2.59	2.79	2.88	2.97	3.03

x	2.8	3.0	3.2	3.4	3.6	3.8	4.0
$f(x)$	2.95	3.09	3.16	3.23	3.41	3.52	3.57

Solution:

(a) Since the larger subinterval is length 2 we move from x_2 to the right by 2×0.38. So $x_4 \approx 2.24$.

(b) The three points that bracket the minimum are x_1, x_4, and x_2.

10 pts. 8. Show that Newton's method converges with order two.

Solution:
Newton's method is:

$$x_{k+1} = x_k - f(x_k)/f'(x_k). \tag{1}$$

Expanding f around x_k by Taylor's formula gives

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{f''(\eta)}{2}(x - x_k)^2,$$

where η is some value between x and x_k. Assuming Newton's method converges to x^*, setting $x = x^*$ in the above equation yields

$$0 = f(x_k) + f'(x_k)(x^* - x_k) + \frac{f''(\eta)}{2}(x^* - x_k)^2,$$

where we have used the fact that $f(x^*) = 0$. Dividing the above equation by $f'(x_k)$ and rearranging gives

$$-\frac{f(x_k)}{f'(x_k)} = (x^* - x_k) + \frac{f''(\eta)}{2f'(x_k)}(x^* - x_k)^2.$$

Using the above to substitute into (1) gives

$$x_{k+1} = x_k + (x^* - x_k) + \frac{f''(\eta)}{2f'(x_k)}(x^* - x_k)^2,$$

which implies

$$|x_{k+1} - x^*| = \left|\frac{f''(\eta)}{2f'(x_k)}\right| |x^* - x_k|^2,$$

and

$$\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} = \lim_{k\to\infty} \left|\frac{f''(\eta)}{2f'(x_k)}\right| = \left|\frac{f''(x^*)}{2f'(x^*)}\right| = \lambda.$$

Since λ is finite (provided $f'(x^*) \neq 0$) we conclude that the order of convergence of Newton's method is (at least) two. (It will be larger than two in the case that $f''(x^*) = 0$.)