**From:** Robert Sharpley <sharpley@math.sc.edu>
**Subject:** **Homeworks #6**
**Date:** February 22, 2006 9:09:53 AM EST
**Cc:** Robert Sharpley <sharpley@math.sc.edu>

---

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%       Homework #5  - Solutions          %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Here is a matlab code for the bisection method that was used in the homework solutions.

```
function [c] = bisection(a,b,n_max, eps)
    u = f(a);
    v = f(b);
    n = 0;
    if (u*v>=0)
        error, 'there might be no root on this interval\n';
    else
        for n=0:n_max
                c = (a+b)/2;
                if (abs(b-a)<eps)
                    fprintf('Convergence: root is between %6.4f and %6.4f\n', a,b);
                    break;
                end;
                w = f(c);
                fprintf('Iteration %3d: f(%6.4f) = %6.4f\n',n,c,w);
                if (w*u == 0)
                    fprintf('Root found: f(%6.4f) = %6.4f\n',c,w);
                    break;
                else
                    if (w*u<0)
                        b = c;
                        v = w;
                    else
                        a = c;
                        u = w;
                    end
                end
        end
    end

% #3.1.1, Page 101
function y = f(x)
    y =  exp(x)-3*x;

% #C3.1.1, , Page 103
function y = f(x)
    y =  x^3-2*x+1-x^2   % or even better by Horner scheme y = (((x-1)*x-2)*x+1
```

-----------------------------------------------------------------

```
% # 3.1.1
ezplot('exp(x)');
```

```
hold on
ezplot('3*x');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = bisection(0,1,20,0.0001);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Iteration   0: f(0.5000) = 0.1487
Iteration   1: f(0.7500) = -0.1330
Iteration   2: f(0.6250) = -0.0068
Iteration   3: f(0.5625) = 0.0676
Iteration   4: f(0.5938) = 0.0295
Iteration   5: f(0.6094) = 0.0112
Iteration   6: f(0.6172) = 0.0021
Iteration   7: f(0.6211) = -0.0023
Iteration   8: f(0.6191) = -0.0001
Iteration   9: f(0.6182) = 0.0010
Iteration  10: f(0.6187) = 0.0005
Iteration  11: f(0.6189) = 0.0002
Iteration  12: f(0.6190) = 0.0000
Iteration  13: f(0.6191) = -0.0000
Convergence: root is between 0.6190 and 0.6191

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = bisection(1,2,20,0.0001);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Iteration   0: f(1.5000) = -0.0183
Iteration   1: f(1.7500) = 0.5046
Iteration   2: f(1.6250) = 0.2034
Iteration   3: f(1.5625) = 0.0832
Iteration   4: f(1.5312) = 0.0302
Iteration   5: f(1.5156) = 0.0054
Iteration   6: f(1.5078) = -0.0066
Iteration   7: f(1.5117) = -0.0006
Iteration   8: f(1.5137) = 0.0024
Iteration   9: f(1.5127) = 0.0009
Iteration  10: f(1.5122) = 0.0001
Iteration  11: f(1.5120) = -0.0003
Iteration  12: f(1.5121) = -0.0001
Iteration  13: f(1.5121) = 0.0000
Convergence: root is between 1.5121 and 1.5121

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% #3.1.3, Page 101, See last page
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = 10:0.5:500;
y = sin(x);
z = 100*atan(x)-50*pi;
plot(x,y,x,z);

% #3.1.8
By applying the formula on page 100 using the logarithms consistently:
  (log(1.0-0.1)-log(2*0.5*1e-8))/log(2)

  ans =
```

```
   26.4234
```

number of iterations is integer, so n>27

```
% #C3.1.1, Page 103
% First approximate the intervals where the root would be - should be three of them , since the degree of the polynomial is three.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = bisection(-2,-1,20,0.00001);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Iteration   0: f(-1.5000) = -1.6250
Iteration   1: f(-1.2500) = -0.0156
Iteration   2: f(-1.1250) = 0.5605
Iteration   3: f(-1.1875) = 0.2903
Iteration   4: f(-1.2188) = 0.1419
Iteration   5: f(-1.2344) = 0.0643
Iteration   6: f(-1.2422) = 0.0246
Iteration   7: f(-1.2461) = 0.0046
Iteration   8: f(-1.2480) = -0.0055
Iteration   9: f(-1.2471) = -0.0005
Iteration  10: f(-1.2466) = 0.0021
Iteration  11: f(-1.2468) = 0.0008
Iteration  12: f(-1.2469) = 0.0002
Iteration  13: f(-1.2470) = -0.0002
Iteration  14: f(-1.2470) = 0.0000
Iteration  15: f(-1.2470) = -0.0001
Iteration  16: f(-1.2470) = -0.0000
Convergence: root is between -1.2470 and -1.2470
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = bisection(0,1,20,0.00001);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Iteration   0: f(0.5000) = -0.1250
Iteration   1: f(0.2500) = 0.4531
Iteration   2: f(0.3750) = 0.1621
Iteration   3: f(0.4375) = 0.0173
Iteration   4: f(0.4688) = -0.0542
Iteration   5: f(0.4531) = -0.0185
Iteration   6: f(0.4453) = -0.0006
Iteration   7: f(0.4414) = 0.0084
Iteration   8: f(0.4434) = 0.0039
Iteration   9: f(0.4443) = 0.0016
Iteration  10: f(0.4448) = 0.0005
Iteration  11: f(0.4451) = -0.0001
Iteration  12: f(0.4449) = 0.0002
Iteration  13: f(0.4450) = 0.0001
Iteration  14: f(0.4450) = 0.0000
Iteration  15: f(0.4451) = -0.0000
Iteration  16: f(0.4450) = -0.0000
Convergence: root is between 0.4450 and 0.4450
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = bisection(1,2,20,0.00001);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Iteration   0: f(1.5000) = -0.8750
Iteration   1: f(1.7500) = -0.2031
Iteration   2: f(1.8750) = 0.3262
Iteration   3: f(1.8125) = 0.0442
Iteration   4: f(1.7812) = -0.0837
Iteration   5: f(1.7969) = -0.0208
Iteration   6: f(1.8047) = 0.0114
Iteration   7: f(1.8008) = -0.0048
Iteration   8: f(1.8027) = 0.0033
Iteration   9: f(1.8018) = -0.0007
Iteration  10: f(1.8022) = 0.0013
Iteration  11: f(1.8020) = 0.0003
Iteration  12: f(1.8019) = -0.0002
Iteration  13: f(1.8019) = 0.0000
Iteration  14: f(1.8019) = -0.0001
Iteration  15: f(1.8019) = -0.0000
Iteration  16: f(1.8019) = -0.0000
Convergence: root is between 1.8019 and 1.8019
```

% #C3.1.3, Page 103

This is a trick problem. Those who can graph by hand on paper won't make a mistake - tangent function is not continuous on [1,2]. There is an infinite jump at pi/2 ~1.57. Thus, method of bisection is not applicable.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Problems #1 and #9, on Page 117 are written on the next to last page.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% #3.2.4, Page 117

```
function x = newtons(x, n_max, delta, eps)
    w = f(x);
    for n=0:n_max
        fp = df(x);
        if (abs(fp)<delta)
                fprintf('Near root, x = %6.4f',x);
                break
        end
        d = w/fp;
        x = x-d;
        w = f(x);
        fprintf('Iteration %3d: f(%6.4f) = %6.4f\n',n,x,w);
        if (abs(d)<eps)
           fprintf('Convergence: root is about %6.4f\n', x);
           break;
        end
    end

function y = f(x)
    y = ((x+2)*x-7)*x*x+3;

function y = df(x)
    y = ((4*x+6)*x-14)*x;
```

```
------------------------------------------------------------

x = newtons(1,20,0.0001,0.0001)
Iteration   0: f(0.7500) = 0.2227
Iteration   1: f(0.7909) = 0.0018
Iteration   2: f(0.7913) = 0.0000
Iteration   3: f(0.7913) = 0.0000
Convergence: root is about 0.7913

x =

    0.7913

x = newtons(2,20,0.0001,0.0001)
Iteration   0: f(1.7500) = 1.6602
Iteration   1: f(1.6416) = 0.2458
Iteration   2: f(1.6190) = 0.0096
Iteration   3: f(1.6180) = 0.0000
Iteration   4: f(1.6180) = 0.0000
Convergence: root is about 1.6180

x =

    1.6180

% #3.2.6, Page 117
x = newtons(2,30,0.0001,1e-45)
Iteration   0: f(1.8750000000) = 0.3436089158
Iteration   1: f(1.7656250000) = 0.1180670870
Iteration   2: f(1.6699218750) = 0.0405689038
Iteration   3: f(1.5861816406) = 0.0139398370
Iteration   4: f(1.5129089355) = 0.0047898523
Iteration   5: f(1.4487953186) = 0.0016458360
Iteration   6: f(1.3926959038) = 0.0005655239
Iteration   7: f(1.3436089158) = 0.0001943191
Iteration   8: f(1.3006578013) = 0.0000667698
Iteration   9: f(1.2630755762) = 0.0000229427
Iteration  10: f(1.2301911291) = 0.0000078833
Iteration  11: f(1.2014172380) = 0.0000027088
Iteration  12: f(1.1762400833) = 0.0000009308
Iteration  13: f(1.1542100728) = 0.0000003198
Iteration  14: f(1.1349338137) = 0.0000001099
Iteration  15: f(1.1180670870) = 0.0000000378
Iteration  16: f(1.1033087011) = 0.0000000130
Iteration  17: f(1.0903951135) = 0.0000000045
Iteration  18: f(1.0790957243) = 0.0000000015
Iteration  19: f(1.0692087588) = 0.0000000005
Iteration  20: f(1.0605576639) = 0.0000000002
Iteration  21: f(1.0529879559) = 0.0000000001
Iteration  22: f(1.0463644614) = 0.0000000000
Iteration  23: f(1.0405689038) = 0.0000000000
Iteration  24: f(1.0354977908) = 0.0000000000
Iteration  25: f(1.0310605669) = 0.0000000000
```

```
Iteration  26: f(1.0271779961) = 0.0000000000
Iteration  27: f(1.0237807466) = 0.0000000000
Iteration  28: f(1.0208081532) = 0.0000000000
Iteration  29: f(1.0182071341) = 0.0000000000
Iteration  30: f(1.0159312423) = 0.0000000000

x =

    1.016
```

The sluggishness of convergence is easily explained if you read the paragraph carefully. On page 111 it says: around a multiple zero Newton's method converges only linearly, not quadratically. If we modify method by incorporating the multiplicity where needed, the convergence becomes amazing - one iteration needed regardless of the point choice! Of course, this effect is due to special choice of a function - polynomial.

Homework solutions #3

3.2.1 Apply the formula for Newtons method:

$f(x) = x^2 - R$ (looking for the solution of an equation $x^2 = R$, or $x^2 - R = 0$)

$f'(x) = 2x$

Thus, $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - R}{2x_n} =$

$= \frac{2x_n^2 - x_n^2 + R}{2x_n} = \frac{x_n^2 + R}{2x_n} = \frac{1}{2}\left(x_n + \frac{R}{x_n}\right).$

What many of you did was to plug $x$ for both $x_n$ and $x_{n+1}$, which would emulate the convergence of the iterations. Unsurprisingly, $x^2 = R$.

3.2.9.

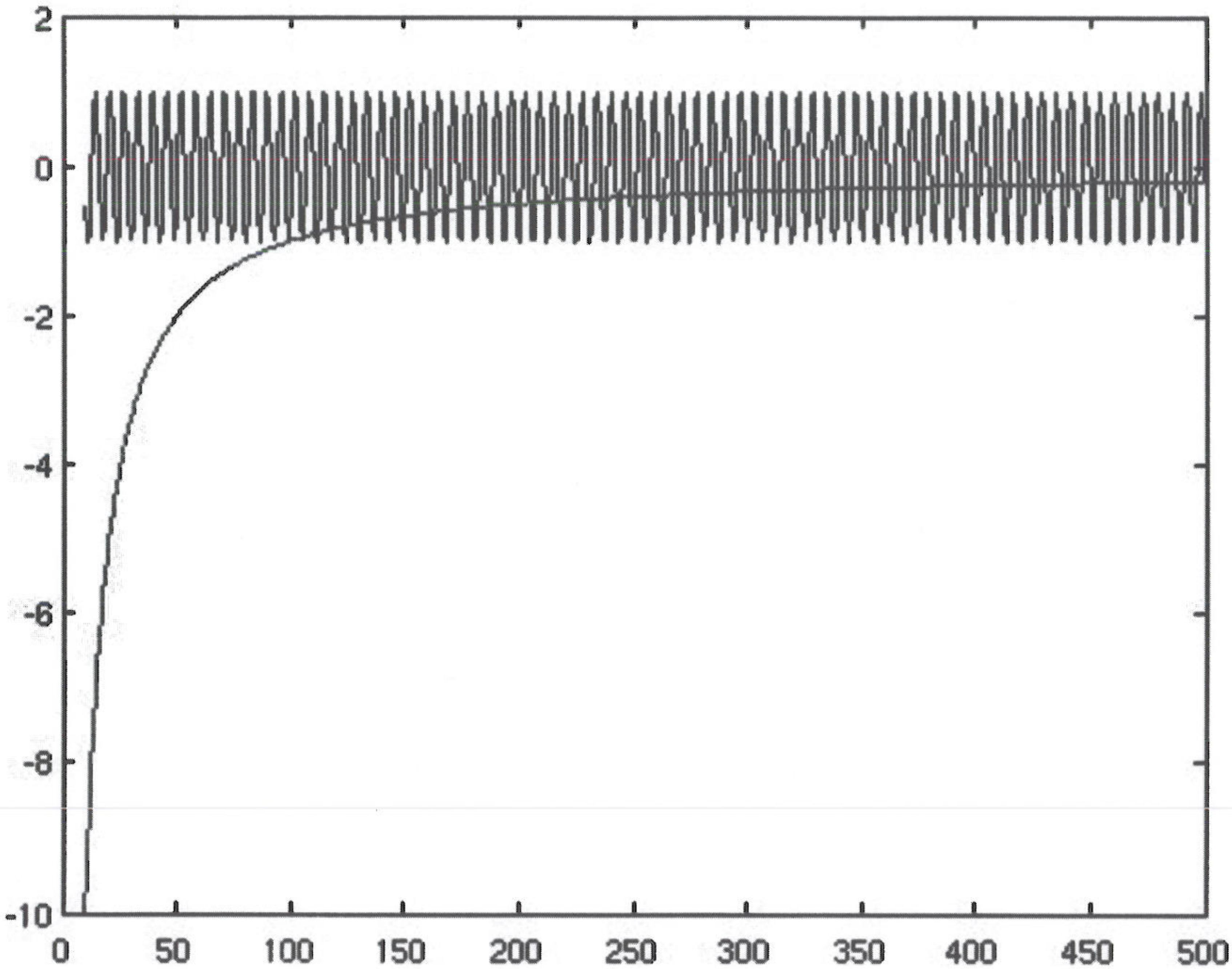
$x_{n+1} = x_n - \tan x_n$. Here the thing to notice is

$x_{n+1} = x_n - \frac{\sin x_n}{\cos x_n}$, where $\cos x = (\sin x)'$, i.e.

we have $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ for $f(x) = \sin x$.

Then the sequence converges to the solution of an equation $f(x) = 0$. $\sin x = 0$ gives us $x = \pi$ (considering roots around $x_0 = 3$).

Simple check of several first terms supports the conclusion.


2
0
-2
-4
-6
-8
-10
0
50
100
150
200
250
300
350
400
450
500