MATH 5610 HW2 SOLUTIONS

Problem 1 see attached code and output.

Problem 2 (B&F 2.2.19)

$$x_n = \frac{1}{2} x_{n-1} + \frac{1}{x_{n-1}}$$
$$= F(x_{n-1})$$
with $F(x) = \frac{1}{2}x + \frac{1}{x} = \frac{x^2+2}{2x}$

a) $F'(x) = \frac{1}{2} - \frac{1}{x^2}$

- F maps $[\sqrt{2}, \infty)$ to $[\sqrt{2}, \infty)$ because:

  $F'(x) \geq 0$ for $x \geq \sqrt{2} \Rightarrow F(x)$ is increasing

  $F'(\sqrt{2}) = 0 \Rightarrow F$ has a min at $x = \sqrt{2}$ with $F(\sqrt{2}) = \sqrt{2}$.

- F is a contraction on $[\sqrt{2}, \infty)$ because:

  $x \geq \sqrt{2} \Rightarrow 0 \leq \frac{1}{x^2} \leq \frac{1}{2} \Rightarrow 0 \leq F'(x) \leq \frac{1}{2} < 1$

$\Rightarrow$ Thm 2.4 guarantees fixed point iter. $x_{n+1} = F(x_n)$ converges for all $x_0 > \sqrt{2}$ to $x_* = F(x_*)$, w/ $x_* = \sqrt{2}$.

b) Let $x_0 \in (0, \sqrt{2})$ then:

$$x_1 = F(x_0) = \frac{x_0^2+2}{2x_0}$$

Since $(x_0 - \sqrt{2})^2 = x_0^2 - 2\sqrt{2}\,x_0 + 2 > 0$ we have

$$x_1 = \frac{x_0^2+2}{2x_0} > \sqrt{2} \Rightarrow \text{method converges by part a).}$$

c) We have shown in a) that method converges for $x_0 \geq \sqrt{2}$ and in b) for $0 < x_0 < \sqrt{2}$, thus method must converge for any $x_0 > 0$.

*) Iteration can be obtained by applying Newton's method to

$$f(x) = x^2 + 2$$

Indeed:

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

Problem 3 See attached code and output

Problem 4 $x = \sqrt{p + \sqrt{p + \sqrt{p + \cdots}}}$

One can view x as the limit (if it exists) of the iteration:

$$x_{n+1} = \sqrt{x_n + p} = F(x_n)$$

③

If this iteration has a limit then:

$$x_* = \sqrt{x_* + p} \Rightarrow x_*^2 - x_* - p = 0$$

$$\Rightarrow x_* = \frac{1 \pm \sqrt{1+4p}}{2}$$

The only root that makes sense in this context is the positive one:

$$x_* = \frac{1+\sqrt{1+4p}}{2} \qquad (\, p=1 \text{ we get golden ratio!})$$

<u>Does this iteration have a limit</u>? The easiest way to show this is by invoking the contractive mapping theorem.

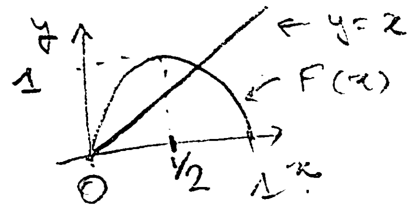
$$F'(x) = \frac{1}{2\sqrt{p+x}} \leqslant \frac{1}{2} < 1 \quad \text{for} \quad x+p \geqslant 1$$
$$x \geqslant 1-p$$

Moreover if $x \in [1, \infty)$ we have:

- $F'(x) \leqslant \frac{1}{2} < 1$
- $F(x) = \sqrt{p+x} \geqslant 1$ (since $p>0$), thus $F(x)$ maps $[1,\infty)$ onto itself.

$\Rightarrow$ By contractive mapping theorem, iteration converges for any $x_0 \geqslant 1$.

<u>Problem 5</u> $F(x) = 4(1-x)x$



- <u>F maps $[0,1]$ to $[0,1]$</u>:

  let $x \in [0,1]$ then:

  $$\left.\begin{array}{l} x \geq 0 \\ 1-x \geq 0 \end{array}\right\} \Rightarrow F(x) \geq 0$$

  Also $F(x)$ has a max when $F'(x) = 4(1-2x) = 0$

  i.e. $x = \frac{1}{2} \Rightarrow F(x) \leq F(\frac{1}{2}) \leq 1$

  then $\forall x \in [0,1]$, $F(x) \in [0,1]$.

- <u>F is not a contraction on $[0,1]$</u>

  Indeed if $x \in [0, \frac{1}{4}]$,

  $$\underbrace{4(1-\tfrac{1}{2})}_{2} \leq F'(x) \leq \underbrace{4(1-0)}_{4}$$

  $$\Rightarrow |F'(x)| \geq 2.$$

- <u>F has a fixed point</u> : $x = F(x)$ is a quadratic:

  $$4(1-x)x - x = 0$$

  $\Leftrightarrow$ $x(3-4x) = 0$ $\Rightarrow$ $x = 0$ or $x = \frac{3}{4}$ are fixed pts of F.

This does not contradict FP theorem, because F is a contraction is only a <u>sufficient</u> condition for having a fixed point.

That is, there could be functions w/ fixed points that are not contractions (such as this example).

Problem 6 (B&F 2.4.12)

We need to prove theorem 2.12:

The function $f \in C^m[a,b]$ has a zero of multiplicity m iff

$$0 = f(p) = f'(p) = \cdots = f^{(m-1)}(p)$$

but $f^{(m)}(p) \neq 0$.

proof:

$\Rightarrow$ Assume f has a zero of multiplicity m at p

then:

$$f(x) = (x-p)^m q(x), \text{ with } \lim_{x \to p} q(x) \neq 0$$

Since $f(x) \in C^m[a,b]$, we must have $q \in C^m[a,b]$ as well.

$$f^{(k)}(p) = \lim_{x \to p} f^{(k)}(x)$$

Write:

$$f^{(k)}(x) = \left((x-p)^m q(x)\right)^{(k)}$$
$$= \sum_{i=0}^{k} \binom{k}{i} \left((x-p)^m\right)^{(i)} q^{(k-i)}(x)$$

Also: $$\left((x-p)^m\right)^{(k)} = \begin{cases} \frac{m!}{(m-k)!}(x-p)^{m-k}, & 0 \le k \le m \\ 0, & k > m \end{cases}$$

$$\Rightarrow \left((x-p)^m\right)^{(k)}\Big|_{x=p} = \begin{cases} 0 & \text{if } 0 \le k \le m-1 \\ m! & \text{if } k = m \\ 0 & \text{if } k \ge m+1 \end{cases}$$

So $f^{(k)}(p) = 0$ for $0 \le k \le m-1$

but $f^{(m)}(p) = m!\, q(p) \neq 0$.

$\underline{\Leftarrow}$ Assume $f(p) = f'(p) = \cdots = f^{(m-1)}(p) = 0$

but $f^{(m)}(p) \neq 0$.

Using Taylor's theorem:

$$f(x) = \underbrace{\sum_{k=0}^{m-1} \frac{1}{k!} f^{(k)}(p)(x-p)^k}_{=0 \text{ by hyp.}} + \frac{1}{m!}(x-p)^m f^{(m)}(\xi(x))$$

Here $\xi(x)$ is a number between x and p.

Clearly $\xi(x) \to p$ as $x \to p$. Thus using that $f^{(m)}$ is continuous:

$$\lim_{x \to p} f^{(m)}(\xi(x)) = f^{(m)}\left(\lim_{x \to p} \xi(x)\right)$$

$$= f^{(m)}(p) \neq 0.$$

We can thus write:

$$f(x) = (x-p)^m \, q(x) \text{ where } q(x) = \frac{1}{m!} f^{(m)}(\xi(x))$$

and $\lim_{x \to p} q(x) \neq 0.$

QED

```
% Bisection method to find root of function f
% in the interval [a,b] to within tol tolerance
%
% inputs:
%  f      function handle
%  a      left bound of interval
%  b      right bound of interval
%  delta  desired tolerance for the root
%  eps    desired tolerance for f(root)
%  maxit  maximum number of iterations
%
% outputs:
%  x      iterate history. The best approx found for the root is x(end)
%  a,b    interval containing the root
function [x,a,b] = bisection(f,a,b,delta,eps,maxit)

 e = b-a; u = f(a); v = f(b);

 % check if we can apply bisection method
 if sign(u)*sign(v)>0
  fprintf('bisection: interval probably does not contain a root\n');
  fprintf('bisection: please refine interval\n');
  return;
 end;

 c = a + e; % midpoint
 x = [c];   % save this iterate
 for k = 1:maxit,
  e = e/2;
  c = a+e; x = [x c]; % save iterate
  w = f(c);
  if (abs(e)<delta || abs(w)<eps)
   % we are happy with this solution
   return;
  end;
  % prepare next iteration
  if (sign(w)*sign(u)<0)
   b=c; v=w; % root is in left half interval
  else
   a=c; u=w; % root is in right half interval
  end;
 end;
```



```
% Newton method to find a root of a function f starting at x0
%
% Inputs
%  f       function we want to find the root of
%  fprime  derivative of f
%  x0      initial iterate
%  tol     desired tolerance or precision
%  maxit   maximum  number of iterations
%
% Outputs
%  xs      iteration history, last iterate is xs(end)
function xs = newton(f,fprime,x0,tol,maxit)

xs = [x0]; % store iterates

x = x0;

for k=1:maxit,
 xnew = x - f(x)/fprime(x);
 xs = [xs xnew];                % store iterates
 if abs(xnew-x)<tol break; end; % check convergence
 x = xnew;                      % prepare next iteration
end;
```

```
% Secant method to find a roof of a function f with initial guesses
% x0 and x1
%
% Inputs
%  f       function we want to find the root of
%  x0      initial iterate
%  x1      initial iterate
%  tol     desired tolerance or precision
%  maxit   maximum  number of iterations
%
% Outputs
%  xs      iteration history, last iterate is xs(end)

function xs = secant(f,x0,x1,tol,maxit)

xs = [x0 x1]; % store iterates

for k=1:maxit,
 xnew = x1 - f(x1)*(x1-x0)/(f(x1)-f(x0));
 xs = [xs xnew];                  % store iterates
 if abs(xnew-x1)<tol break; end; % check convergence
 x0=x1; x1=xnew;                  % prepare next iteration
end;
```



```
% B&F 2.1.6 try out the bisection method in Matlab

tol = 1e-5; maxit=30;

fprintf('***** problem B&F 2.1.6 c\n');
f = @(x) x^2 - 4*x +4 - log(x);

x1=bisect(f,1,2,tol,tol,maxit);
x2=bisect(f,2,4,tol,tol,maxit);

% display results
for i=1:max([length(x1),length(x2)]),
 if (i<=length(x1) && i <= length(x2) )
  fprintf('%13.8f %13.8f\n',x1(i),x2(i));
 else
  if (i>length(x1))
   fprintf('%13s %13.8f\n',' ',x2(i));
  else
   fprintf('%13.8f %13s\n',x1(i),' ');
  end;
 end;
end;


fprintf('***** problem B&F 2.1.6 d\n');
f = @(x) x + 1 - 2*sin(pi*x);
x1=bisect(f,0,0.5,tol,tol,maxit);
x2=bisect(f,0.5,1,tol,tol,maxit);

% display results
for i=1:max([length(x1),length(x2)]),
 if (i<=length(x1) && i <= length(x2) )
  fprintf('%13.8f %13.8f\n',x1(i),x2(i));
 else
  if (i>length(x1))
   fprintf('%13s %13.8f\n',' ',x2(i));
  else
   fprintf('%13.8f %13s\n',x1(i),' ');
  end;
 end;
end;
```

```
% HW 2 Problem 3
tol = 1e-5; maxit=20;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf('**** B&F 2.3.6 a and 2.3.8 a\n')


f      = @(x) exp(x) + 2^-x + 2*cos(x) - 6;
fprime = @(x) exp(x) - log(2)*2^-x - 2*sin(x);

% compute root with Newton's and Secant methods
xnewt = newton(f,fprime,2,tol,maxit);
xsec  = secant(f,1,2,tol,maxit);

% display results
fprintf('%13s %13s\n','newton','secant');
for i=1:max([length(xnewt),length(xsec)]),
 if (i≤length(xnewt) ∧∧ i ≤ length(xsec) )
  fprintf('%13.8f %13.8f\n',xnewt(i),xsec(i));
 else
  if (i>length(xnewt))
   fprintf('%13s %13.8f\n',' ',xsec(i));
  else
   fprintf('%13.8f %13s\n',xnewt(i),' ');
  end;
 end;
end;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf('**** B&F 2.3.6 b and 2.3.8 b\n')
f      = @(x) log(x-1) + cos(x-1);
fprime = @(x) 1/(x-1) + sin(x-1);

% compute root with Newton's and Secant methods
xnewt = newton(f,fprime,1.3,tol,maxit);
xsec  = secant(f,1.3,1.4,tol,maxit);

% display results
fprintf('%13s %13s\n','newton','secant');
for i=1:max([length(xnewt),length(xsec)]),
 if (i≤length(xnewt) ∧∧ i ≤ length(xsec) )
  fprintf('%13.8f %13.8f\n',xnewt(i),xsec(i));
 else
  if (i>length(xnewt))
   fprintf('%13s %13.8f\n',' ',xsec(i));
  else
   fprintf('%13.8f %13s\n',xnewt(i),' ');
  end;
 end;
end;
```



```
>> prob1
***** problem B∧F 2.1.6 c
   2.00000000    4.00000000
   1.50000000    3.00000000
   1.25000000    3.50000000
   1.37500000    3.25000000
   1.43750000    3.12500000
   1.40625000    3.06250000
   1.42187500    3.03125000
   1.41406250    3.04687500
   1.41015625    3.05468750
   1.41210938    3.05859375
   1.41308594    3.05664062
   1.41259766    3.05761719
   1.41235352    3.05712891
   1.41247559    3.05688477
   1.41241455    3.05700684
   1.41238403    3.05706787
   1.41239929    3.05709839
   1.41239166
***** problem B∧F 2.1.6 d
   0.50000000    1.00000000
   0.25000000    0.75000000
   0.12500000    0.62500000
   0.18750000    0.68750000
   0.21875000    0.65625000
   0.20312500    0.67187500
   0.21093750    0.67968750
   0.20703125    0.68359375
   0.20507812    0.68164062
   0.20605469    0.68261719
   0.20556641    0.68212891
   0.20581055    0.68188477
   0.20593262    0.68200684
   0.20599365    0.68194580
   0.20602417    0.68197632
   0.20603943
   0.20603180

>> prob3
**** B∧F 2.3.6 a and 2.3.8 a
       newton        secant
   2.00000000    1.00000000
   1.85052134    2.00000000
   1.82975120    1.67830848
   1.82938372    1.80810288
   1.82938360    1.83229846
                 1.82933117
                 1.82938347
                 1.82938360
**** B∧F 2.3.6 b and 2.3.8 b
       newton        secant
   1.30000000    1.30000000
   1.36851649    1.40000000
   1.38979527    1.39811755
   1.39561671    1.39774706
   1.39717876    1.39774848
   1.39759633
   1.39770785
   1.39773763
   1.39774558
>>
```