Moed bet exam 5770 − my answers to Matlab questions
Note there are many ways to do everything!

%%

Section A, Question 1

```
      N=pi/2*[0 1 ; 2 0 ]

N =
         0    1.5708
    3.1416         0

      [sin(N);cos(N)]

ans =

         0    1.0000
    0.0000         0
    1.0000    0.0000
   -1.0000    1.0000

     [sin(N)*(cos(N).*N),(sin(N)*cos(N)).*N]

ans =

   -3.1416         0         0    1.5708
         0    0.0000    0.0000         0
```

%%

Section A, Question 2

For example make M-file q2.m

```
==============================================================
function z=q2(p)

  z=-quad(@(x) (1+p*cos(x))./(1+p^2*x.^2+x.^4/4), -100, 100);
  % because of the x^4 factor in the denominator taking -100..100 is enough!

end
===============================================================
```

Then try

```
[a b]=fminsearch( @q2, 0.7)

a =

    0.6400


b =

   -3.5722
```

note this is − the value of the function at the maximum

see plot of the function on the website

%%

Section A, Question 3

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

Make M-file q3.m

=====================================
function z=q3(x,y)

  z=log(x.^2+cos(y)^2).*(x.^2+y^2<1);

end
=====================================

Note this works for vector x and scalar y!

Then do

dblquad(@q3,-1,1,-1,1)

ans =

   -0.0943

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Section B, Question 2
---------------------
a) M-file b2a.m
============================
function q=b2a(p)

% the polynomial is p(1)x^(n-1) + p(2)x^(n-2) + ... + p(n-1)x + p(n) = 0
% the derivative is (n-1)p(1)x^(n-2) + (n-2)p(2)x^(n-3) + ... + p(n-1) = 0

n=length(p);
q=zeros(n-1,1);
for i=1:(n-1)
        q(i)=(n-i)*p(i);
end
============================

b) M-file b2b.m
============================
function [maxs mins]=b2b(p)

d=length(p)-1; % the degree of the polynomial
fd=b2a(p);     % the derivative polynomial
sd=b2a(fd);    % the second derivative polynomial
r=roots(fd);   % the critical points - including complex ones
maxs=[];
mins=[];
for i=1:length(r)
   if imag(r(i))==0     % if have a real root
      % compute the second derivative at the root - many other options
        val_sd=sum(sd.* ( (r(i)*ones(d-1,1)).^(((d-2):-1:0)') ) );
      if val_sd<0
         maxs=[maxs;r(i)];
      elseif val_sd>0
         mins=[mins;r(i)];
      end
   end
end
% this won't work if there is a minimum or maximum with second deriv zero
============================

c) M-file b2c.m
============================
```

```
function depths=b2c(p)

[maxs mins]=b2b(p);
n=length(mins);                  %  number of minima
depths=zeros(size(mins));        %  this vector will hold the depths

for i=1:n

    % for each minimum find the nearest max on the left and on the right

   left=-Inf;
   right=Inf;
   for j=1:size(maxs)
        if maxs(j)<mins(i) && maxs(j)>left
           left=maxs(j);
        elseif maxs(j)>mins(i) && maxs(j)<right
           right=maxs(j);
        end
   end

   if left==-Inf && right==Inf
           depths(i)=Inf;
   elseif left==-Inf
       depths(i)=polyval(p,right)-polyval(p,mins(i));
   elseif right==Inf
       depths(i)=polyval(p,left)-polyval(p,mins(i));
   else
       depths(i)=min( polyval(p,right)-polyval(p,mins(i)) , polyval(p,left)-polyval(p,mins(i)));
   end

end
============================

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Section B, Question 3
---------------------
a) M-file b3a.m
============================
function S=b3a(A)

% make the eigenvalues and eigenvectors
[V D]=eig(A);

% count the real eigenvalues
n=0;
for i=1:size(A,1)
    if imag(D(i,i))==0
        n=n+1;
    end
end

% sort the real eigenvalues and eigenvectors
realD=zeros(n,1);
realV=zeros(size(A,1),n);
j=1;
for i=1:size(A,1)
    if imag(D(i,i))==0
       realD(j)=D(i,i);
       realV(:,j)=V(:,i);
       j=j+1;
    end
end

% do the sum
S=0;
```

```
for i=1:n
    for j=(i+1):n
        c=dot(realV(:,i),realV(:,j))/norm(realV(:,i))/norm(realV(:,j));
        S=S+realD(i)*realD(j)*c^2;
    end
end
============================
```

in fact the use of "norm" is unnecessary, as matlab chooses eigenvectors with norm 1.

b) M-file b3b.m − starts exactly the same as previous function

```
============================
function [minangle maxangle]=b3b(A)

% make the eigenvalues and eigenvectors
[V D]=eig(A);

% count the real eigenvalues
n=0;
for i=1:size(A,1)
    if imag(D(i,i))==0
        n=n+1;
    end
end

% sort the real eigenvalues and eigenvectors
realD=zeros(n,1);
realV=zeros(size(A,1),n);
j=1;
for i=1:size(A,1)
    if imag(D(i,i))==0
       realD(j)=D(i,i);
       realV(:,j)=V(:,i);
       j=j+1;
    end
end

% find the minimum and maximum
minangle=pi/2;
maxangle=0;
for i=1:n
    for j=(i+1):n
        c=abs(dot(realV(:,i),realV(:,j))/norm(realV(:,i)))/norm(realV(:,j));
        angle=acos(c);
        if angle>maxangle
            maxangle=angle;
        end
        if angle<minangle
            minangle=angle;
        end
    end
end
============================
```

%%