Answers to Problem Set 2
Group name: Ferienspass

Sebastian Kühnl: 5642348
Alexander Dück (as: reebyte): 5504077
Patrick (as: paddyblank):
Christian:

Question 1

1.1

```matlab
function [ r ] = mybisection( fun, a, b, maxiter,
    eps_step, eps_abs )
  % Finding a root using the bisection algorithm
  % fun    := function
  % a, b   := two points in the domain
  % maxiter := maximum number of iterations before the
      algorithm breaks


  % Check requirements:
  if (maxiter<=0) || (eps_step<0) || (eps_abs<0)
    error('input parameters invalid');
  end
  %% fun at a or b already zero?
  if ( fun(a) == 0 )
    r = a;
    return;
  elseif ( fun(b) == 0 )
    r = b;
    return;
  %% [a,b] does contain a root?
  elseif ( fun(a) * fun(b) > 0 )
    error( 'f(a) and f(b) do not have opposite signs' );
  end

  % Iteration
  for i = 1:maxiter
    c = (a + b)/2; % midpoint
    % Is c already a root?

    % Determine the interval to be considered
    if ( fun(c) == 0 )
```

```
31        r = c;
32        return;
33     elseif ( fun(c)*fun(a) < 0 )
34        % proceed with [a, c] if f(a) and f(c) have
                 opposite signs
35        b = c;
36     else
37        % proceed with [c, b] because f(b) and f(c) then
                 have opposite signs
38        a = c;
39     end
40
41     if ( b-a < eps_step )
42        if ( abs( fun(a) ) < abs( fun(b) ) && abs( fun(a) )
                 < eps_abs )
43          r = a;
44          return;
45        elseif ( abs( fun(b) ) < eps_abs )
46          r = b;
47          return;
48        end
49     end
50   end
51
52   error( 'No root could be found: the bisection did not
          converge' );
53 end
```

1.2

```
1 %PS2 P1
2 %2. first function
3 function [y]=ffunction(t)
4      y=t.^3+4-1./t;
5 end
```

```
1 %PS2 P1
2 %2. second function
3 function [z]=fffunction(v)
4      z=-exp(-v)+exp(-v.^2);
5 end
```

```
1 clear;
2 %PS2 P1 2.
3 %get an idea how the function looks like, such that
       values can be guessed
```

```matlab
4   i =1:300;
5   g=ffunction(i-40/40);
6   h=fffunction(i-40);
7   figure
8   plot(g);
9   title('Function 1');
10  figure
11  plot(h,'--r');
12  title('Function 2');
13  %do bisection
14  maxiter=20;
15  e=0.0001;
16  d=0.0001;
17  x_l=-5;
18  x_h=0.7;
19  fun=@ffunction;
20  a=mybisection(fun,x_l,x_h,maxiter,e,d);
21  disp(a)        %-1.6632
22  x_l=0.01;
23  x_h=0.7;
24  b=mybisection(fun,x_l,x_h,maxiter,e,d);
25  disp(b)        %0.2490
26  %x_l=0.25;
27  %x_h=100;
28  %l=mybisection(fun,x_l,x_h,maxiter,e,d);
29  %disp(l)     %(error: not opposite signs)
30  %x_l=-50;
31  %x_h=-1.7;
32  %l=mybisection(fun,x_l,x_h,maxiter,e,d);
33  %disp(l)     %(error: not opposite signs) --> no more
        zeros
34  %function 1 has zeros at -1.6632 and 0.2490
35  %function 2
36  x_l=0.1;
37  x_h=100;
38  fan=@fffunction;
39  c=mybisection(fan,x_l,x_h,maxiter,e,d);
40  disp(c)        %1
41  x_l=0;
42  x_h=100;
43  t=mybisection(fan,x_l,x_h,maxiter,e,d);
44  disp(t)        %0
45  %x_l=-0.1;
46  %x_h=-20;
47  %s=mybisection(fan,x_l,x_h,maxiter,e,d);
48  %disp(s)     (error: not opposite signs)
```

```
49  %x_l=1.1;
50  %x_h=100;
51  %v=mybisection(fan,x_l,x_h,maxiter,e,d);
52  %disp(v)      (error: not opposite signs)  --> no more
        zeros
53  %function 2 has zeros at 0 and 1
```

1.3

A market equilibrium occurs when markets clear. This implies no excess demand (D) or supply (S) of Goods. Thus, $q_D = q_S$. This only occurs when $p_D = p_S$ (the market clearing price prevails).

$$p_D = p_S$$

using

$$p_D = a - b * q_D \text{ and } p_S = c + d * q_S^\psi$$

we get

$$a - b * q_D = c + d * q_S^\psi$$

$$0 = c + d * q_S^\psi - (a - b * q_D)$$

$$0 = c - a + d * q_S^\psi + b * q_D$$

$$0 = b * q_D + d * q_S^\psi - (a - c)$$

Since $q_D = q_S$ holds, this can be written as

$$0 = b * q + d * q^\psi - (a - c)$$

∎

a=3, b=0.5, c=d=1, ψ=0.5

$$
\begin{aligned}
&0 = 0.5 \cdot q + \sqrt{q} - (3 - 1) \\
&0 = 0.5 \cdot q + \sqrt{q} - 2 && \text{substitute:} x^2 = q \\
&0 = 0.5x^2 + x - 2 && \text{abc-formula} \\
&x_1 = 1.236 && \text{resubstitution: } q = x^2 \\
&x_2 = -3.236 && \text{resubstitution invalid} \\
&\to q = \pm 1.112 && q > 0 \\
&\to q* = 1.112 && \text{insert into: } p = 3 - 0.5q \\
&\to p* = 2.444
\end{aligned}
$$

```matlab
function p=demand(k)
    a=3;
    b=0.5;
    p=a-b.*k;
end
```

```matlab
function p=supply(l)
c=1;
d=c;
psi=0.5;
p=c+d.*l.^psi;
end
```

```matlab
%Univariate Function Problem 1.3
function dif=difference(q)
a=3;
b=0.5;
c=1;
d=c;
psi=0.5;
dif=b.*q+d.*q.^psi-a+c;
end
```

```matlab
clear;
%PS2 Problem 1 3.
%get an idea how the difference looks like
x=0:100;
plot(x,difference(x));  %only 1 zero in this interval!
%Use bisec algorithm
fun=@difference;
maxiter=20;
a=3;
b=0.5;
psi=0.5;
c=1;
d=c;
x_l=0;
x_h=5;
q=mybisection(fun,x_l,x_h,maxiter,0.0001,0.0001);
disp(q);          %q=1.5279
p=demand(q);
disp(p);          %p=2.2361
%use fzero with guess 1
z=fzero(fun,1);
disp(z);          %q=1.5279
pp=demand(z);
```

5

```matlab
24  disp(pp);              %p=2.2361
25  %Gauss-Seidel fixed point iteration
26  %initial values
27  i=1;
28  q(i)=0.1;
29  p(i)=supply(q(i));
30  qdiff=1; %just >epsilon to get into the loop
31  %stopping criterion
32  e=0.00001;
33  delta=0.001;
34  maxiter=25;
35  %start with q_supply=0.1
36  while (i-1<maxiter) && (qdiff>e)
37     i=i+1;
38     q(i)=(p(i-1)-a)/(-b);
39     p(i)=supply(q(i));
40     qdiff=abs(q(i)-q(i-1));
41  end
42  d=abs(supply(q(i))-p(i)); %difference in prices
43  if ((i-1)==maxiter) || (d>delta) || (isnan(d))
44      disp('failure')
45  else
46      disp('success')
47  end
48  disp(i-1);             %#iterations = 25
49  disp(q(i))            %q=1.5405
50  disp(p(i))            %p=2.2412
51  %it displays 'failure' --> convergence not fast enough
        --> reorder system of equations
52  p=[];
53  q=[];
54  i=1;
55  p(i)=0.1;
56  q(i)=((p(i)-c)/d)^(1/psi);
57  qdiff=1; %just >epsilon to get into the loop
58  %start with p_supply=0.1
59  while (i-1<maxiter) && (qdiff>e)
60     i=i+1;
61     p(i)=demand(q(i-1));
62     q(i)=((p(i)-c)/d)^(1/psi);
63     qdiff=abs(q(i)-q(i-1));
64  end
65  d=abs(supply(q(i))-p(i));  %difference in prices
66  if ((i-1)==maxiter) || (d>delta) || (isnan(d))
67      disp('failure')
68  else
```

```matlab
69       disp('success')
70  end
71  disp(i-1);              %#iterations = 1
72  disp(q(i))              %q= +inf
73  disp(p(i))              %p= -inf
74  %failure --> fast divergence --> try dampening
75  %dampening (of first method)
76  lambda=0.75;
77  p=[];
78  q=[];
79  i=1;
80  q(i)=0.1;
81  qtilde(i)=0;
82  p(i)=supply(q(i));
83  qdiff=1; %just >epsilon to get into the loop
84  %stopping criterion
85  e=0.0001;
86  delta=0.0001;
87  maxiter=25;
88  %start with q_supply=0.1
89  while (i-1<maxiter) && (qdiff>e)
90      i=i+1;
91      qtilde(i)=(p(i-1)-a)/(-b);                %new value
             without dampening
92      q(i)=lambda*qtilde(i)+(1-lambda)*q(i-1); %new value
             with dampening
93      p(i)=supply(q(i));
94      qdiff=abs(q(i)-q(i-1));
95  end
96  d=abs(supply(q(i))-p(i)); %difference in prices
97  if ((i-1)==maxiter) || (d>delta) || (isnan(d))
98      disp('failure')
99  else
100     disp('success')
101 end
102 disp(i-1);             %#iterations = 12
103 disp(q(i))            %q=1.5279
104 disp(p(i))            %p=2.2361
105 %'success', same solutions as mybisection and fzero
106 %dampening (of second method)
107 p=[];
108 q=[];
109 lambda=1.5;
110 i=1;
111 p(i)=0.1;
112 ptilde(i)=0;
```

```matlab
113  q( i )=((p( i )-c )/d)^(1/ psi ) ;
114  q d i f f =1; %just >epsilon  to  get  into  the  loop
115  %start  with  p_supply=0.1
116  while  ( i -1<maxiter )  &&  ( q d i f f >e)
117      i=i +1;
118      p t i l d e ( i )=demand ( q ( i -1));
119      p( i )=lambda∗ p t i l d e ( i )+(1−lambda )∗p( i -1);
120      q( i )=((p( i )-c )/d)^(1/ psi ) ;
121      q d i f f=abs ( q ( i )-q( i -1));
122  end
123  d=abs ( supply ( q ( i ))-p( i ));   %difference  in  quantities
124  if  (( i -1)==maxiter )  ||  (d>delta )  ||  ( isnan (d))
125      disp ( ' failure ')
126  else
127      disp ( ' failure ')
128  end
129  disp ( i -1);              %#iterations = 1
130  disp ( q ( i ))            %q= Inf
131  disp (p( i ))             %p= -Inf
132  %' failure '    something  must  be  wrong  here  ( reordering ,
          etc . . . )
```

Question 2

```matlab
1   clear ;
2   i =1;
3   data=xlsread ( 'MRW92QJE-data . xls ');
4   %every  row  has  1 NaN,  which  is  fine
5   while  i<=length ( data )
6       summe=0;
7       for  j =1: size ( data ,2)
8           if  isnan ( data ( i , j ))
9               summe=summe+1;
10          end
11      end
12      if  summe>1
13          data ( i ,:) =[];
14          i=i -1;
15      end
16      i=i +1;
17  end
18  %subsamples
19  nonoil =[];
20  interm =[];
```

```matlab
21  oecd = [ ] ;
22  for i =1:length (data)
23      if data(i,3)==1
24          nonoil=[nonoil;data(i,:)];
25      end
26      if data(i,4)==1
27          interm=[interm;data(i,:)];
28      end
29      if data(i,5)==1
30          oecd=[oecd;data(i,:)];
31      end
32  end
33  %multiple regression
34  %nonoil
35  y1=log(nonoil(:,7))-log(nonoil(:,6));
36  x1=[ones(length(nonoil),1) log(nonoil(:,6)) log(nonoil
        (:,10)) log(nonoil(:,9)+0.05) log(nonoil(:,11))];
37  [b1,a11,a12,a13,stats1]=regress(y1,x1);
38  %interm
39  y2=log(interm(:,7))-log(interm(:,6));
40  x2=[ones(length(interm),1) log(interm(:,6)) log(interm
        (:,10)) log(interm(:,9)+0.05) log(interm(:,11))];
41  [b2,a21,a22,a23,stats2]=regress(y2,x2);
42  %nonoil
43  y3=log(oecd(:,7))-log(oecd(:,6));
44  x3=[ones(length(oecd),1) log(oecd(:,6)) log(oecd(:,10))
        log(oecd(:,9)+0.05) log(oecd(:,11))];
45  [b3,a31,a32,a33,stats3]=regress(y3,x3);
46  %display
47  obs=[length(nonoil) length(interm) length(oecd)];
48  Sample={'Observations:'; 'Constants:' ;'SE'; 'ln(Y60)' ;'
        SE'; 'ln(I/G)'; 'SE'; 'ln(n+g+delta)' ;'SE'; 'ln(
        schooling)'; 'SE'; 'R^2'; 's.e.e'};
49  Nonoil=[length(nonoil) b1(1) abs((a11(1,1)+a11(1,2))*sqrt
        (obs(1))/1.96) b1(2) abs((a11(2,1)+a11(2,2))*sqrt(obs
        (1))/1.96) b1(3) abs((a11(3,1)+a11(3,2))*sqrt(obs(1))
        /1.96) b1(4) abs((a11(4,1)+a11(4,2))*sqrt(obs(1))
        /1.96) b1(5) abs((a11(5,1)+a11(5,2))*sqrt(obs(1))
        /1.96) stats1(1) stats1(4)];
50  Nonoil=transpose(Nonoil);
51  Intermediate=[length(interm) b2(1) abs((a21(1,1)+a21(1,2)
        )*sqrt(obs(2))/1.96) b2(2) abs((a21(2,1)+a21(2,2))*
        sqrt(obs(2))/1.96) b2(3) abs((a21(3,1)+a21(3,2))*sqrt(
        obs(2))/1.96) b2(4) abs((a21(4,1)+a21(4,2))*sqrt(obs
        (2))/1.96) b2(5) abs((a21(5,1)+a21(5,2))*sqrt(obs(2))
        /1.96) stats2(1) stats2(4)];
```

```
52  Intermediate=transpose(Intermediate);
53  OECD=[length(oecd) b3(1) abs((a31(1,1)+a31(1,2))*sqrt(obs
        (3))/1.96) b3(2) abs((a31(2,1)+a31(2,2))*sqrt(obs(3))
        /1.96) b3(3) abs((a31(3,1)+a31(3,2))*sqrt(obs(3))
        /1.96) b3(4) abs((a31(4,1)+a31(4,2))*sqrt(obs(3))
        /1.96) b3(5) abs((a31(5,1)+a31(5,2))*sqrt(obs(3))
        /1.96) stats3(1) stats3(4)];
54  OECD=transpose(OECD);
55  T=table(Sample,Nonoil,Intermediate,OECD)
56  T.Properties.VariableNames{'Sample'} = 'Sample';
57  T.Properties.VariableNames{'Nonoil'} = 'Nonoil';
58  T.Properties.VariableNames{'Intermediate'} = '
        Intermediate';
59  T.Properties.VariableNames{'OECD'} = 'OECD';
```

T =

Sample	Nonoil	Intermediate	OECD
'Observations:'	98	75	22
'Constants:'	1.0639	1.5984	2.4839
'SE'	10.747	14.125	11.888
'ln(Y60)'	-0.29933	-0.37604	-0.4009
'SE'	3.0237	3.3231	1.9188
'ln(I/G)'	0.52091	0.5364	0.37022
'SE'	5.262	4.7402	1.7719
'ln(n+g+delta)'	-0.14027	-0.14316	-0.14139
'SE'	1.4169	1.2651	0.67672
'ln(schooling)'	0.234	0.27431	0.23588
'SE'	2.3637	2.424	1.1289
'R^2'	0.49145	0.47226	0.70221
's.e.e'	0.1057	0.091475	0.022794