Answers to Problem Set 2
Group name: Ferienspass

Sebastian Kühnl: 5642348
Alexander Dück (as: reebyte): 5504077
Patrick Blank (as: paddyblank): 6729110
Christian Wierschem: 6729288

Question 1

1.1

```
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 )
```

```
      r = c;
      return;
    elseif ( fun(c)*fun(a) < 0 )
      % proceed with [a, c] if f(a) and f(c) have opposite signs
      b = c;
    else
      % proceed with [c, b] because f(b) and f(c) then have opposite signs
      a = c;
    end

    if ( b-a < eps_step )
      if ( abs( fun(a) ) < abs( fun(b) ) && abs( fun(a) ) < eps_abs )
        r = a;
        return;
      elseif ( abs( fun(b) ) < eps_abs )
        r = b;
        return;
      end
    end
  end

  error( 'No root could be found: the bisection did not converge' );
end
```

1.2

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

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

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

```
i=1:300;
g=ffunction(i-40/40);
h=fffunction(i-40);
figure
plot(g);
title('Function 1');
figure
plot(h,'--r');
title('Function 2');
%do bisection
maxiter=20;
e=0.0001;
d=0.0001;
x_l=-5;
x_h=0.7;
fun=@ffunction;
a=mybisection(fun,x_l,x_h,maxiter,e,d);
disp(a)     %-1.6632
x_l=0.01;
x_h=0.7;
b=mybisection(fun,x_l,x_h,maxiter,e,d);
disp(b)     %0.2490
%x_l=0.25;
%x_h=100;
%l=mybisection(fun,x_l,x_h,maxiter,e,d);
%disp(l)    %(error: not opposite signs)
%x_l=-50;
%x_h=-1.7;
%l=mybisection(fun,x_l,x_h,maxiter,e,d);
%disp(l)    %(error: not opposite signs) --> no more zeros
%function 1 has zeros at -1.6632 and 0.2490
%function 2
x_l=0.1;
x_h=100;
fan=@fffunction;
c=mybisection(fan,x_l,x_h,maxiter,e,d);
disp(c)    %1
x_l=0;
x_h=100;
t=mybisection(fan,x_l,x_h,maxiter,e,d);
disp(t)    %0
%x_l=-0.1;
%x_h=-20;
%s=mybisection(fan,x_l,x_h,maxiter,e,d);
%disp(s)    (error: not opposite signs)
```

```
%x_l=1.1;
%x_h=100;
%v=mybisection(fan,x_l,x_h,maxiter,e,d);
%disp(v)     (error: not opposite signs)  --> no more zeros
%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

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

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

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

```
%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
```

```
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);
```

```
disp(pp);        %p=2.2361
%Gauss-Seidel fixed point iteration
%initial values
i=1;
q(i)=0.1;
p(i)=supply(q(i));
qdiff=1; %just >epsilon to get into the loop
%stopping criterion
e=0.00001;
delta=0.001;
maxiter=25;
%start with q_supply=0.1
while (i-1<maxiter) && (qdiff>e)
   i=i+1;
   q(i)=(p(i-1)-a)/(-b);
   p(i)=supply(q(i));
   qdiff=abs(q(i)-q(i-1));
end
d=abs(supply(q(i))-p(i)); %difference in prices
if ((i-1)==maxiter) || (d>delta) || (isnan(d))
     disp('failure')
else
     disp('success')
end
disp(i-1);           %#iterations = 25
disp(q(i))           %q=1.5405
disp(p(i))           %p=2.2412
%it displays 'failure' --> convergence not fast enough --> reorder system of equations
p=[];
q=[];
i=1;
p(i)=0.1;
q(i)=((p(i)-c)/d)^(1/psi);
qdiff=1; %just >epsilon to get into the loop
%start with p_supply=0.1
while (i-1<maxiter) && (qdiff>e)
   i=i+1;
   p(i)=demand(q(i-1));
   q(i)=((p(i)-c)/d)^(1/psi);
   qdiff=abs(q(i)-q(i-1));
end
d=abs(supply(q(i))-p(i));  %difference in prices
if ((i-1)==maxiter) || (d>delta) || (isnan(d))
     disp('failure')
else
```

```
    disp('success')
end
disp(i-1);          %#iterations = 1
disp(q(i))          %q= +inf
disp(p(i))          %p= -inf
%failure --> fast divergence --> try dampening
%dampening (of first method)
lambda=0.75;
p=[];
q=[];
i=1;
q(i)=0.1;
qtilde(i)=0;
p(i)=supply(q(i));
qdiff=1; %just >epsilon to get into the loop
%stopping criterion
e=0.0001;
delta=0.0001;
maxiter=25;
%start with q_supply=0.1
while (i-1<maxiter) && (qdiff>e)
  i=i+1;
  qtilde(i)=(p(i-1)-a)/(-b);                %new value without dampening
  q(i)=lambda*qtilde(i)+(1-lambda)*q(i-1); %new value with dampening
  p(i)=supply(q(i));
  qdiff=abs(q(i)-q(i-1));
end
d=abs(supply(q(i))-p(i)); %difference in prices
if ((i-1)==maxiter) || (d>delta) || (isnan(d))
    disp('failure')
else
    disp('success')
end
disp(i-1);          %#iterations = 12
disp(q(i))          %q=1.5279
disp(p(i))          %p=2.2361
%'success', same solutions as mybisection and fzero
%dampening (of second method)
p=[];
q=[];
lambda=1.5;
i=1;
p(i)=0.1;
ptilde(i)=0;
```

```
q(i)=((p(i)-c)/d)^(1/psi);
qdiff=1; %just >epsilon to get into the loop
%start with p_supply=0.1
while (i-1<maxiter) && (qdiff>e)
   i=i+1;
   ptilde(i)=demand(q(i-1));
   p(i)=lambda*ptilde(i)+(1-lambda)*p(i-1);
   q(i)=((p(i)-c)/d)^(1/psi);
   qdiff=abs(q(i)-q(i-1));
end
d=abs(supply(q(i))-p(i));  %difference in quantities
if ((i-1)==maxiter) || (d>delta) || (isnan(d))
      disp('failure')
else
      disp('failure')
end
disp(i-1);           %#iterations = 1
disp(q(i))           %q= Inf
disp(p(i))           %p= -Inf
%'failure'   something must be wrong here (reordering,
    etc...)
```

Question 2

```
clear;
i=1;
data=xlsread('MRW92QJE-data.xls');
%every row has 1 NaN, which is fine
while i<=length(data)
     summe=0;
     for j=1:size(data,2)
          if isnan(data(i,j))
               summe=summe+1;
          end
     end
     if summe>1
          data(i,:)=[];
          i=i-1;
     end
     i=i+1;
end
%subsamples
nonoil=[];
interm=[];
```

```
oecd=[];
for i=1:length(data)
    if data(i,3)==1
        nonoil=[nonoil;data(i,:)];
    end
    if data(i,4)==1
        interm=[interm;data(i,:)];
    end
    if data(i,5)==1
        oecd=[oecd;data(i,:)];
    end
end
%multiple regression
%nonoil
y1=log(nonoil(:,7))-log(nonoil(:,6));
x1=[ones(length(nonoil),1) log(nonoil(:,6)) log(nonoil
    (:,10)) log(nonoil(:,9)+0.05) log(nonoil(:,11))];
[b1,a11,a12,a13,stats1]=regress(y1,x1);
%interm
y2=log(interm(:,7))-log(interm(:,6));
x2=[ones(length(interm),1) log(interm(:,6)) log(interm
    (:,10)) log(interm(:,9)+0.05) log(interm(:,11))];
[b2,a21,a22,a23,stats2]=regress(y2,x2);
%nonoil
y3=log(oecd(:,7))-log(oecd(:,6));
x3=[ones(length(oecd),1) log(oecd(:,6)) log(oecd(:,10))
    log(oecd(:,9)+0.05) log(oecd(:,11))];
[b3,a31,a32,a33,stats3]=regress(y3,x3);
%display
obs=[length(nonoil) length(interm) length(oecd)];
Sample={'Observations:'; 'Constants:' ;'SE'; 'ln(Y60)' ;'
    SE'; 'ln(I/G)'; 'SE'; 'ln(n+g+delta)' ;'SE'; 'ln(
    schooling)'; 'SE'; 'R^2'; 's.e.e'};
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)];
Nonoil=transpose(Nonoil);
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)];
```

```
Intermediate=transpose(Intermediate);
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)];
OECD=transpose(OECD);
T=table(Sample,Nonoil,Intermediate,OECD)
T.Properties.VariableNames{'Sample'} = 'Sample';
T.Properties.VariableNames{'Nonoil'} = 'Nonoil';
T.Properties.VariableNames{'Intermediate'} = '
    Intermediate';
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