University of Helsinki / Department of Mathematics and Statistics

SCIENTIFIC COMPUTING

Exercise 11 / Solutions

1. The Gram–Schmidt method also applies to orthogonalize a system of functions. Use this method to orthogonalize the system $\{1, x, x^2, x^4\}$. of the space $C([-1,1])$ with the inner product $(f,g) = \int_{-1}^{1} f(x)g(x)\,dx$.

Solution: Write $p_0(x) = 1$, $p_1(x) = x$, $p_2(x) = x^2$, $p_3(x) = x^4$ and orthogonalize the set $\{p_i\}_{i=0}^3$ in the sense of the L^2 norm. In other words, we form the set $\{q_i\}_{i=0}^3$, that spans the same set as $\{p_i\}_{i=0}^3$ and which also satisfied

$$(q_i, q_j) = \int_{-1}^{1} q_i(x) q_j(x)\,dx = 0\,,$$

when $i, j = 0, \dots, 3$ and $i \neq j$. Set

$$q_0(x) = 1\,.$$

The Gram–Schmidt algorithm yields

$$q_1(x) = p_1(x) - \frac{(p_1,q_0)}{(q_0,q_0)} q_0(x) \tag{1}$$

$$q_2(x) = p_2(x) - \sum_{k=0}^{1} \frac{(p_2,q_k)}{(q_k,q_k)} q_k(x) \tag{2}$$

$$q_3(x) = p_3(x) - \sum_{k=0}^{2} \frac{(p_3,q_k)}{(q_k,q_k)} q_k(x)\,. \tag{3}$$

Because

$$(p_1, q_0) = \int_{-1}^{1} x\,dx = \Big/_{-1}^{1} \frac{1}{2}x^2 = 0$$

and

$$(q_0, q_0) = \int_{-1}^{1} 1^2\,dx = 2\,,$$

we have

$$p_1(x) = x\,.$$

Further

$$\begin{aligned}
(p_2, q_0) &= \int_{-1}^{1} x^2\,dx = \Big/_{-1}^{1} \frac{1}{3}x^3 = \frac{2}{3}\,,\\
(p_2, q_1) &= \int_{-1}^{1} x^3\,dx = \Big/_{-1}^{1} \frac{1}{4}x^4 = 0\,,\\
(q_1, q_1) &= \int_{-1}^{1} x^2\,dx = \frac{2}{3}\,,
\end{aligned}$$

and hence

$$q_2(x) = x^2 - \frac{2}{3\cdot 2} - 0\cdot x = x^2 - \frac{1}{3}\,.$$

Now we are in a position to compute

$$\begin{aligned}
(p_3, q_0) &= \int_{-1}^{1} x^4\,dx = \Big/_{-1}^{1} \frac{1}{5}x^5 = \frac{2}{5}\,,\\
(p_3, q_1) &= \int_{-1}^{1} x^5 = \Big/_{-1}^{1} \frac{1}{6} = 0\,,\\
(p_3, q_2) &= \int_{-1}^{1} x^4(x^2 - \frac{1}{3})\,dx = \Big/_{-1}^{1} (\frac{1}{7}x^7 - \frac{1}{15}x^5)\\
&= \frac{2}{7} - \frac{2}{15} = \frac{30-14}{105} = \frac{16}{105}\,,\\
(q_2, q_2) &= \int_{-1}^{1} (x^2 - \frac{1}{3})^2\,dx = \int_{-1}^{1} (x^4 - \frac{2}{3}x^2 + \frac{1}{9})\,dx\\
&= \Big/_{-1}^{1} (\frac{1}{5}x^5 - \frac{2}{9}x^3 + \frac{1}{9}x) = \frac{1}{5} - \frac{2}{9} + \frac{1}{9} + \frac{1}{5} - \frac{2}{9} + \frac{1}{9}\\
&= \frac{2}{5} - \frac{2}{9} = \frac{18-10}{45} = \frac{8}{45}\,.
\end{aligned}$$

Substitution into (3) yields

$$\begin{aligned}
q_3(x) &= x^4 - \frac{2}{5\cdot 2} - 0\cdot x - \frac{16}{105}\cdot\frac{45}{8}\cdot(x^2 - \frac{1}{3})\\
&= x^4 - \frac{6}{7}x^2 + \frac{3}{35}\,.
\end{aligned}$$

The orthogonal polynomial q_i are

$$\begin{aligned} q_0(x) &= 1\,,\\ q_1(x) &= x\,,\\ q_2(x) &= x^2-\frac{1}{3}\,,\\ q_3(x) &= x^4-\frac{6}{7}x^2+\frac{3}{35}\,. \end{aligned}$$

2. Part (b) of this problem deals with the so called Gibbs phenomenon.

(a) Show that for each fixed x the number $S_n(x)=\frac{nx}{1+n^2x^2}$ approaches zero when n grows to ∞, and find the extremum values of $S_n(x)$ with respect to x. Graph the function $S_n(x)$ when n=2:2:10.

(b) Show that the Fourier series of $f(x)=(\pi-x)/2$ on $(0,2\pi)$ is $\lim_{n\to\infty} S_n(x)$, where $S_n(x)=\sum_{k=1}^{n}\frac{1}{k}\sin(kx)$. Graph S_n for n=10:2:20 and find graphically the global maximum $x_0\in(0,2\pi)$ of $S_n(x)$ and estimate graphically the number $|S_n(x_0)-f(x_0)|/|f(x_0)|$.

Recall the Fourier series of a continuous function $f:[0,2\pi]\to\mathbf{R}$

$$\frac{1}{2}a_0+\sum_{n=1}^{\infty}(a_n\cos(nx)+b_n\sin(nx)),$$

where

$$a_n=\frac{1}{\pi}\int_0^{2\pi}f(x)\cos(nx)dx,\ n=0,1,2,\ldots$$

and

$$b_n=\frac{1}{\pi}\int_0^{2\pi}f(x)\sin(nx)dx,\ n=1,2,3,\ldots$$

Solution: (a) Because $1/n\to 0$, when $n\to\infty$, it follows for all $x\in\mathbf{R}$ that (the case $x=0$ will be studied separately)

$$S_n(x)=\frac{nx}{1+n^2x^2}=\frac{x}{\frac{1}{n}+nx^2}\to 0\,,$$

for $n\to\infty$. In the same way we see that for each fixed n, $S_n(x)\to 0$, when $x\to\pm\infty$, and hence because S_n is differentiable everywhere its extrema are found in the set of zeros of

$$S_n'(x)=\frac{n(1+n^2x^2)-nx\cdot 2n^2x}{(1+n^2x^2)^2}\,.$$

Therefore for the extemum points we get the condition

$$n^3x^2-2n^3x^2+n=0\quad\Leftrightarrow\quad 1-n^2x^2=0\quad\Leftrightarrow\quad x=\pm\frac{1}{n}\,.$$

The extremum values are

$$S_n(\frac{\pm 1}{n})=\frac{\pm 1}{1+1}=\pm\frac{1}{2}\,.$$

The graph of S_n is plotted in the first picture, when $n=10$. The key point in the function S_n or rather in the sequence $(S_n)_{n=1}^{\infty}$ is the fact that while it tends to zero at each individual point x when n grows with bound, nevertheless the maximum value of S_n is always the same, $\frac{1}{2}$.

(b) The Fourier series of $f(x)=(\pi-x)/2$ on the interval $(0,2\pi)$ is computed now. First,

$$\begin{aligned} a_0 &= \frac{1}{\pi}\int_0^{2\pi}\frac{\pi-x}{2}\cos 0\,dx=\frac{1}{\pi}\Big/_0^{2\pi}\frac{\pi}{2}x-\frac{x^2}{4}\\ &= \frac{1}{\pi}\left(\frac{2\pi^2}{2}-\frac{4\pi^2}{4}\right)=0\,. \end{aligned}$$

The general form of the cosine terms is

$$\begin{aligned} a_n &= \frac{1}{\pi}\int_0^{2\pi}\frac{\pi-x}{2}\cos nx\,dx\\ &= \frac{1}{\pi}\Big/_0^{2\pi}\frac{\pi-x}{2}\frac{1}{n}\sin nx-\frac{1}{\pi}\int_0^{2\pi}\left(-\frac{1}{2}\right)\frac{1}{n}\sin nx\,dx\\ &= \frac{1}{\pi}\Big/_0^{2\pi}\frac{1}{2n^2}(-\cos nx)=\frac{1}{2\pi n^2}(-1-(-1))=0\,. \end{aligned}$$

Thus all the cosine terms vanish. This could have deduced also directly bu noticing that the function f satisfies $f(-x)=-f(x)$. The coefficients of the remaining sin terms are

$$\begin{aligned} b_n &= \frac{1}{\pi}\int_0^{2\pi}\frac{\pi-x}{2}\sin nx\,dx\\ &= -\frac{1}{\pi}\Big/_0^{2\pi}\frac{\pi-x}{2}\frac{1}{n}\cos nx+\frac{1}{\pi}\int_0^{2\pi}\left(-\frac{1}{2}\right)\frac{1}{n}\cos nx\,dx\\ &= -\frac{1}{\pi n}\left(\frac{\pi-2\pi}{2}-\frac{\pi}{2}\right)-\frac{1}{\pi}\Big/_0^{2\pi}\frac{1}{2n^2}\sin nx\\ &= \frac{2\pi}{2\pi n}=\frac{1}{n}\,. \end{aligned}$$

Therefore the Fourier series of f is

$$f(x) = \sum_{n=1}^{\infty} \frac{1}{n} \sin nx \,.$$

The twentieth partial sum of the Fourier series is drawn in the other picture. Because the Fourier series is periodic with the period 2π but

$$\lim_{x\to 0+} f(x) = \frac{\pi}{2} \neq \frac{-\pi}{2} = \lim_{x\to 2\pi-} f(x) \,,$$

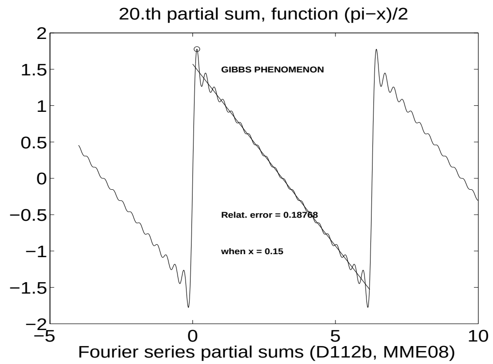
we see that at the points $n2\pi$, with $n = 0, \pm 1, \pm 2, \ldots$, the Fourier series has a strong jump discontinuity. Because of the nature of the Fourier approximation on both sides of this point there will be a clear peak of "balanching oscillation", which is clearly seen even when n is growing and the approximation farther away the jump discontinuity is improving. The emergence of this peak is called the Gibbs phenomenon and it is typical for the Fourier approximation in connection with points of discontinuity. According to the other picture the heigth of the peak for the twentieth partial sum is about 19 % of the value of the function at the point (and the same approxomate results is obtained also if we refined the subdivision).

```
% FILE  d112.m  begins.
close all;
clear
x = -5: 0.01: 5;
figure(1)
axes('FontSize',[15],'FontWeight','bold');
for n=2:4:10
  sn = n*x./(1 + (n*n)*x.^2);
  plot(x,sn), title(['Function number ' num2str(n) ])
  xlabel('D112a, MME08'), grid, pause(1.5);
end
figure(2)
axes('FontSize',[20],'FontWeight','bold');
x = -4: 0.01: 10;   x4f = 0: 0.1: 2*pi;
one = ones(1,length(x));
for n=10:2:20
  osasumma = 0*one;
  for k=1:n
    osasumma = osasumma + (1/k)*sin(k*x);
  end
  [osmax,j] = max(osasumma( (0<x)&(x<pi) ));
  j = j + length( x(x<=0) );
```



```
  ero = (osmax/(0.5*(pi-x(j)))-1);
  plot(x4f,(pi-x4f)/2, x,osasumma, x(j),osasumma(j),'ko')
  title([num2str(n) '.th partial sum, function (pi-x)/2'])
  xlabel(' Fourier series partial sums (D112b, MME08)')
  text(1,-0.5,['Relat. error = ' num2str(ero)],...
       'FontWeight','bold')  % Needed in spite of startup settings
  text(1,-1,['when x = ' num2str(x(j))],'FontWeight','bold')
  text(1,1.5,'GIBBS PHENOMENON','FontWeight','bold'), pause(1.5)
end
% FILE  d112.m  ends.
```

Output:



3. Find the nth partial sum of the Fourier series of $f(x) = x^2$ on $(0 < x < 2\pi)$ and graph it for n=4:2:10.

Solution: We compute the Fouriercoefficients of $f(x) = x^2$ when $x \in (0, 2\pi)$.

$$\begin{aligned} a_0 &= \frac{1}{\pi}\int_0^{2\pi} x^2 \cos 0 \, dx = \frac{1}{\pi}\int_0^{2\pi} x^2 \, dx = \frac{1}{\pi}\Big/_0^{2\pi} \frac{1}{3}x^3 \\ &= \frac{1}{\pi}\cdot\frac{1}{3}\cdot 8\pi^3 = \frac{8}{3}\pi^2 \,, \end{aligned}$$

$$\begin{aligned}
a_n &= \frac{1}{\pi}\int_0^{2\pi} x^2 \cos nx\, dx \\
&= \frac{1}{\pi}\Big/_0^{2\pi} x^2 \frac{1}{n}\sin nx - \frac{1}{\pi}\int_0^{2\pi} 2x \cdot \frac{1}{n}\sin nx\, dx \\
&= \frac{1}{\pi}\cdot 4\pi^2 \frac{1}{n}\sin 2\pi n + \frac{1}{\pi}\Big/_0^{2\pi} 2x \cdot \frac{1}{n^2}\cos nx - \frac{1}{\pi}\int_0^{2\pi} 2\cdot\frac{1}{n^2}\cos nx\, dx \\
&= \frac{4\pi}{n}\sin 2\pi n + \frac{1}{\pi}4\pi\frac{1}{n^2}\cos 2\pi n - \frac{1}{\pi}\Big/_0^{2\pi} 2\frac{1}{n^3}\sin nx \\
&= \frac{4}{n^2} - \frac{1}{\pi}\left(\frac{2}{n^3}\sin 2\pi n - \frac{2}{n^3}\sin 0\right) \\
&= \frac{4}{n^2} \qquad (n = 1, 2, 3, \ldots),
\end{aligned}$$

$$\begin{aligned}
b_n &= \frac{1}{\pi}\int_0^{2\pi} x^2 \sin nx\, dx \\
&= -\frac{1}{\pi}\Big/_0^{2\pi} x^2 \frac{1}{n}\cos nx + \frac{1}{\pi}\int_0^{2\pi} 2x\frac{1}{n}\cos nx\, dx \\
&= -\frac{1}{\pi}4\pi^2\frac{1}{n}\cos 2\pi n + \frac{1}{\pi}\Big/_0^{2\pi} 2x\frac{1}{n^2}\sin nx - \frac{1}{\pi}\int_0^{2\pi}\frac{2}{n^2}\sin nx\, dx \\
&= -\frac{4\pi}{n} + \frac{1}{\pi}\frac{4\pi}{n^2}\sin 2\pi n + \frac{1}{\pi}\Big/_0^{2\pi}\frac{2}{n^3}\cos nx \\
&= -\frac{4\pi}{n} + \frac{1}{\pi}\left(\frac{2}{n^3}\cos 2\pi n - \frac{2}{n^3}\cos 0\right) \\
&= -\frac{4\pi}{n} + \frac{2}{n^3\pi} - \frac{2}{n^3\pi} = -\frac{4\pi}{n} \qquad (n = 1, 2, 3, \ldots).
\end{aligned}$$

The Fourier series of the function $f(x) = x^2$ is therefore

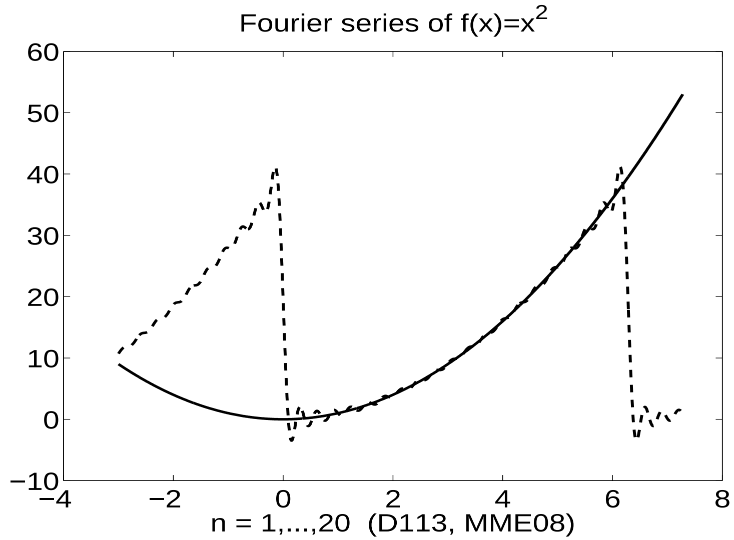
$$x^2 = \frac{4}{3}\pi^2 + \sum_{n=1}^{\infty}\left(\frac{4}{n^2}\cos nx - \frac{4\pi}{n}\sin nx\right) .$$

```
% FILE  d113.m  begins.
clf, clear
x = -3:0.01:2*pi+1;
one = ones(size(x));
y = x.^2;
```



```
figure(1)
axes('FontSize',[20],'FontWeight','bold');

for nmax = 4:2:20
  z = (8*(pi^2)*one)/6;
  for n = 1:nmax
    z = z + 4*cos(n*x)/(n^2) - 4*pi*sin(n*x)/n;
  end
  plt=plot(x,y,'-',x,z,'--');
  set(plt,'LineWidth',2);
  title('Fourier series of f(x)=x^2')
  xlabel(['n = 1,...,' num2str(nmax) '  (D113, MME08)'])
  pause(1.5)
end

% FILE  d113.m  ends.
```

Output:



4. Consider again the problem of fitting a "line with a break point" to a data set, as in d101. Now, instead of choosing the break point (s, t) with a mouse click as we did in d101, use the method of the program parfit to find the best break point $(s, t) = (\lambda_1, \lambda_2)$. The object function will be, with the notation of the solution to d101, s1+s2. Apply this optimized version of d101 to the data of d101. Recall that the object function value obtained in d101, after the fitting was 2.62. Do you get a better value this time?

Solution:

```
% FILE  d114old.m  begins.
% Fitting with a break point
% USES: fitbrf.m, fitbrobj.m  widemarg.m
% MME04
% Fit a broken line into xdata,ydata
% Both broken lines go through the point (lam(1),lam(2))
% and the left line is obtained by the LSQ-fit to
% xdata(j), ydata(j) xdata(j)<lam(1)
% the right one to
% xdata(j), ydata(j), xdata(j)>=lam(1)
% BUT with the constraint that both go through (lam(1),lam(2))
clear global xdata
clear global ydata
global xdata;
global ydata;
%path(path,'../../util')  % widemarg.m

x=-2:0.1:4; y=0.2*sin(3*x);
y(x<1)=y(x<1)+0.5*(x(x<1)-1);
y(x>=1)=y(x>=1)+2*(x(x>=1)-1);

xdata =x;
ydata =y;
lam0=[xdata(10) ydata(10)];        % Initial guess for lambda
y0=fitbrobj( lam0);
                                   % Initial value of object function
lam=fminsearch('fitbrobj',lam0);
                      % lam is the fitted value for
                      % the parameter vector
x1=min(xdata); x2=max(xdata);
dx=0.05*(x2-x1);
x=x1:dx:x2;
yfit=fitbrf(lam, x);
yfinal=fitbrobj(lam);     % Final value of the object function
clf;
axes('FontSize',[15],'FontWeight','bold'); hold on;
```



```
title(['Object function values: start =' num2str(y0) ', final = ' ...
       num2str(yfinal)])
plt=plot(x,yfit,xdata,ydata,'k.','MarkerSize',6);  grid;
txt1=' {\bf Fitted curve (solid)}';
ax=axis;
y1=ax(3); y2=ax(4);
%text(x(5),y1+0.1*(y2-y1),txt1,'FontWeight','bold','FontSize',[20]);
txt2=' {\bf Data point (dots)}';
%text(xdata(8),y1+0.3*(y2-y1),txt2,'FontWeight','bold','FontSize',[20]);
ylabel(['MME04/demo/d10/d103.m'],'FontSize',[16]);
% Observe following: In fitbrf lam(1) was forced between
% min(xdata) and max(xdata)
% Thus the true lam(1) is obtained by the foll. transformation
lam(1)=min(xdata)+(max(xdata)-min(xdata))*(abs(lam(1))/(1+abs(lam(1))));
xlabel(['[x,y] = ' mat2str(lam,4)],'FontWeight','bold','FontSize',[20]);
set(plt,'LineWidth',2.5);
fprintf('lam=');
fprintf(' %10.4f',lam);
fprintf('\nfinal value of object function = %12.4f\n',yfinal);
widemarg(gcf)
% FILE  d114old.m  ends.
```

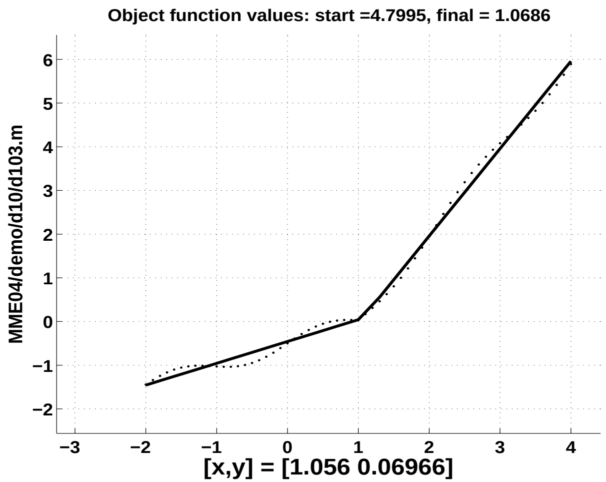
```
function y=fitbrf(lam,x)

global xdata;
global ydata;
x1=min(xdata); x2=max(xdata);
xbrp=x1+(x2-x1)*(abs(lam(1)) /(1+abs(lam(1))));
                              % This forces the break point
                              % between x1 and x2
ybrp=lam(2);
xx= xdata(xdata<=xbrp);
yy= ydata(xdata<=xbrp);
k1=sum((xx-xbrp).*(yy-ybrp));
k1=k1/sum((xx-xbrp).^2);
xx1=x(x<=xbrp);
yy1=ybrp+k1*(xx1-xbrp);
xx= xdata(xdata>=xbrp);
yy= ydata(xdata>=xbrp);
k2=sum((xx-xbrp).*(yy-ybrp));
k2=k2/sum((xx-xbrp).^2);
xx2=x(x>xbrp);
yy2=ybrp+k2*(xx2-xbrp);
y=[yy1 yy2];
```

```
% FILE  fitbrf.m  ends.



function y=fitbrobj(lambda)
global xdata;
global ydata;
y=norm(fitbrf(lambda,xdata)-ydata);
% FILE  fitbrobj.m  ends.
```

Note that the definition `xbrp=x1+(x2-x1)*(abs(lam(1)) /(1+abs(lam(1))));` in fitbrf.m always forces the break point between the extreme values of the data abscissas (the alternative solution d114.m below does not have this feature).

Output:

```
lam=     1.0561     0.0697
final value of object function =       1.0686
```



Alternatively, one may use the following variant which avoids the use of global variables.

```
% FILE  d114.m  begins.
% Fitting with a break point
% MME08
```



```
% Fit a broken line into xdata,ydata
% Both broken lines go through the point (lam(1),lam(2))
% and the left line is obtained by the LSQ-fit to
% xdata(j), ydata(j) xdata(j)<lam(1)
% the right one to
% xdata(j), ydata(j), xdata(j)>=lam(1)
% BUT with the constraint that both go through (lam(1),lam(2))
% TO fit the line y=kx+b through the point (s,t)
% the coefficients k, b are (Problem Set 2):
% num= sum(( xdata-s).*(ydata-t));
% den= sum(( xdata-s).^2);
% k=num/den;  b= t-k*s;

function w=myf

%Synthetic data is generated:
x=-2:0.1:4; y=0.2*sin(3*x);
y(x<1)=y(x<1)+0.5*(x(x<1)-1);
y(x>=1)=y(x>=1)+2*(x(x>=1)-1);
xdata =x; ydata =y;
lam0=[xdata(10) ydata(10)];        % Initial guess for lambda
y0=fitbrf( lam0, xdata, ydata);    % firbrf is the function to be minimized
                                   % Initial value of object function
lam=fminsearch(@fitbrf,lam0,[], xdata,ydata);
                      % lam is the fitted value for
                      % the parameter vector,i.e. the break point
m=sum(xdata<lam(1));  % xdata(j), j<=m is to the left of break point
[k1,b1]=linethrough(lam(1),lam(2), xdata(1:m), ydata(1:m));
[k2,b2]=linethrough(lam(1),lam(2), xdata(m+1:end), ydata(m+1:end));
% Draw picture of fitted polygonal line  yfit
x1=min(xdata); x2=max(xdata);  dx=0.05*(x2-x1); x=x1:dx:x2;
xx1=x(x<lam(1));  xx2=x(x>=lam(1));
yfit=[polyval([k1, b1],xx1)   polyval([k2, b2],xx2) ];
yfinal=fitbrf(lam,xdata,ydata);     % Final value of the object function
clf;
axes('FontSize',[15],'FontWeight','bold'); hold on;
title(['Object function values: start =' num2str(y0) ', final = ' ...
       num2str(yfinal)])
plt=plot(x,yfit,xdata,ydata,'k.','MarkerSize',6);  grid;
txt1=' {\bf Fitted curve (solid)}';
ax=axis;
y1=ax(3); y2=ax(4);
text(x(8),y1+0.1*(y2-y1),txt1,'FontWeight','bold','FontSize',[20]);
```

```
txt2=' {\bf Data point (dots)}';
text(x(3),y1+0.5*(y2-y1),txt2,'FontWeight','bold','FontSize',[20]);
ylabel(['MME08/demo/d11/d114.m'],'FontSize',[16]);
xlabel(['[x,y] = ' mat2str(lam,4)],'FontWeight','bold','FontSize',[20]);
set(plt,'LineWidth',2.5);
fprintf('lam=');
fprintf(' %10.4f',lam);
fprintf('\nInitial/final value of object function = %12.4f  %12.4f\n',...
y0,yfinal);
widemarg(gcf)


function [k, b]=linethrough(s,t,x,y)
% Problem Set 2
num= sum(( x-s).*(y-t));
den= sum(( x-s).^2);
k=num/den;  b= t-k*s;


function w= fitbrf(lam, xdata,ydata)
x=xdata; y=ydata;
s=lam(1); t= lam(2);
m=sum(x <s);
[k1, b1]=linethrough(s,t,x(1:m),y(1:m));
alku=sum((y(1:m)-k1*x(1:m)-b1).^2);
[k2, b2]=linethrough(s,t,x(m+1:end),y(m+1:end));
loppu=sum((y(m+1:end)-k2*x(m+1:end)-b2).^2);
w=alku+loppu;

% FILE  d114.m  ends.
```

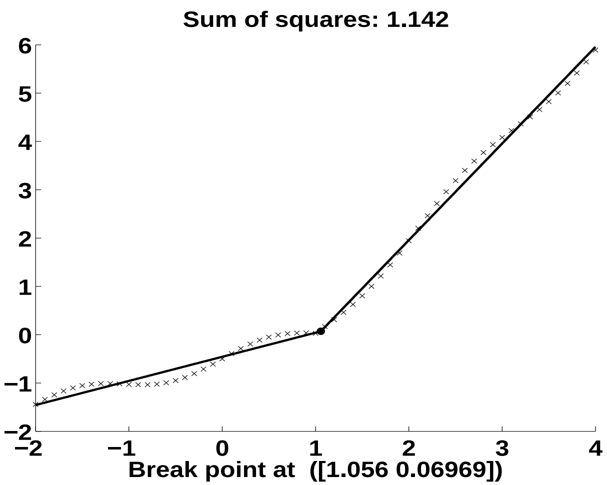
Also, here is an additional solution.

```
%file v11t4.m begin
% author: Lasse Lybeck
function v11t4

% data set, to which the line with a break point should be fit:
x=-2:0.1:4; y=0.2*sin(3*x);
y(x<1)=y(x<1)+0.5*(x(x<1)-1);
y(x>=1)=y(x>=1)+2*(x(x>=1)-1);

% another data set, to test the program:
%x = 0:0.1:10;
```



```
%y = zeros(size(x));
%brx = 10*rand;
%first = x < brx;
%kk = 10*rand(2,1)-5;
%y(first) = kk(1)*x(first) + rand(size(x(first)));
%y(~first) = kk(1)*brx - kk(2)*(x(~first) - brx)  + rand(size(x(~first)));

% initial guess:
br0 = [mean(x) mean(y)];

% find the best break point by minimizing the error:
br = fminsearch(@(br) getError(x,y,br) , br0);

% get the slopes k1 and k2 with the break point:
k = fit(x,y,br);

% plot the lines:
xx = min(x):0.01:max(x);
firstLine = xx < br(1);
xx1 = xx(firstLine);
xx2 = xx(~firstLine);
yy1 = br(2) + k(1)*(xx1-br(1));
yy2 = br(2) + k(2)*(xx2-br(1));

hold on
plot(x,y,'rx')
plot(br(1),br(2),'k.','MarkerSize',20)
plot(xx1,yy1,'b-','LineWidth',2)
plot(xx2,yy2,'b-','LineWidth',2)
title([ 'Sum of squares: ' num2str( getError(x,y,br),4) ])
xlabel(['Break point at  ('  mat2str([br(1),br(2)],4) ')' ]   )
hold off

fprintf('Break point at (%.3f,%.3f)\n',br(1),br(2))
fprintf('Sum of squares: %.5f\n', getError(x,y,br))

end

function error = getError(x,y,br)
% The function 'getError' calculates the error (sum of squares) of the best
% fit lines with a given break point 'br'.

A = @(k,xpts,ypts) sum( (ypts-br(2)-k*(xpts-br(1))).^2 );
```

```
k = fit(x,y,br);
firstPts = x < br(1);
error = A(k(1),x(firstPts),y(firstPts)) + A(k(2),x(~firstPts),y(~firstPts));

end

function k = fit(x,y,br)
% The function 'fit' returns the slopes of the best fit lines through
% the break point 'br'.

k = zeros(2,1);

A = @(k,xpts,ypts) sum( (ypts-br(2)-k*(xpts-br(1))).^2 );

firstPts = x < br(1);
k(1) = fminsearch(@(k) abs(A(k,x(firstPts),y(firstPts))) , 1);
k(2) = fminsearch(@(k) abs(A(k,x(~firstPts),y(~firstPts))) , 1);

end

%file v11t4.m end
```



5. An astronomer has the following observations about a comet approaching the Earth.

Table 1: Comet coordinates

x	1.02	0.95	0.87	0.77	0.67	0.56	0.44	0.30	0.16	0.01
y	0.39	0.32	0.27	0.22	0.18	0.15	0.13	0.12	0.13	0.15

Determine the equation of the comet on the basis of this data using a quadratic function

$$ay^2 + bxy + cx + dy + e = x^2.$$

Hint: The problem yields the overdetermined system

$$ay_i^2 + bx_iy_i + cx_i + dy_i + e = x_i^2 \,, i = 1, ..., 10,$$

which we will solve with the LSQ method for the coefficient vector $sol = (a, b, c, d, e)^T$. We rewrite this as $M * sol = w$ and its solution is obtained with $sol = M\backslash w$ (or, alternatively, $sol = pinv(M) * w$).

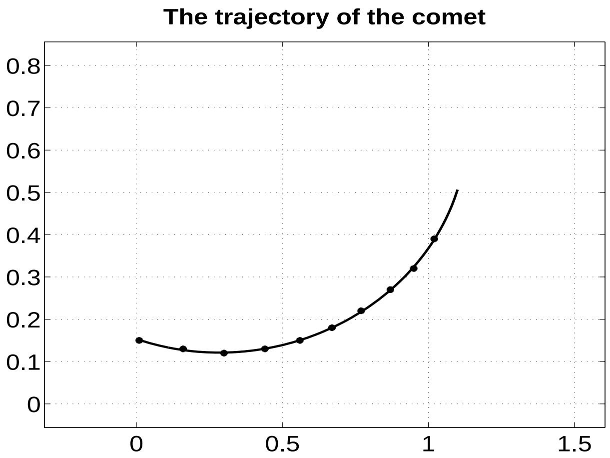
Solution:

```
% FILE d115.m begins.
path(path,'../util')
close all
x=[  1.02  0.95  0.87   0.77  0.67  0.56  0.44 0.30 0.16 0.01];
y=[0.39  0.32  0.27  0.22  0.18  0.15  0.13  0.12  0.13  0.15];


%Determine the equation of the comet on the basis of this data using
%a quadratic function
%$$ay ^2 + bxy + cx +dy + e= x^2.$$
m=[x' y'];
coef=[m(:,2).^2 m(:,1).*m(:,2) m(:,1)  m(:,2) ones(length(x),1)  ];
rhs=m(:,1).^2;
sol=coef\rhs;

a=sol(1); b=sol(2); c=sol(3);
d=sol(4); e=sol(5);
disp('a b c d e =')
disp(sol')
t=0:0.01:1.1;
% Solve the quadratic equation for y:
yt=(-(d+b*t) +sqrt((d+b*t).*(d+b*t) -4*a*(c*t+e-t.^2)))./(2*a);
```

```
figure(1)
axes('FontSize',[20],'FontWeight','bold');
plot(x,y,'*',t,yt,'LineWidth',2)
grid on
title('The trajectory of the comet',...
      'FontSize',[20],'FontWeight','bold');
widemarg(gcf)
% FILE d115.m ends.
```

Output:

```
a b c d e =
   -2.6356    0.1436    0.5514    3.2229   -0.4329
```



6. Write a program numdf, which computes the numerical derivative of a function at the points in a given vector, using the function numder. The program call should be of the form

```
numdf('myf(x)',z, 1e-4)
```

where z = 0:0.05:1;, and myf is a function. Plot the error of the numerical derivation using the command pic('cos(x)- numdf(''sin(x)'', x, 1e-4)'). Hint: The file hlp116.m contains numder and pic.

Solution:


```
% FILE d116.m begins.
% USES: numdf, pic, widemarg
% clear;
function d116
pic('cos(x)- numdf(''sin(x)'', x, 1e-4)');
widemarg(gcf);



function dy = numder(y,h)
%NUMDER gives numerical approximation of derivative
% of equally spaced data y = y(x+j*h), j=1,...,m, m >=5.
% The 5-point rule numerical differentiation coefficients,
% from Abramowitz-Stegun 25.3.6 are used.  To compute
%       dy0=f'(x0) at a single point x0 set e.g. h =0.001
%       x=x0-2*h:h:x0+2*h; y = f(x); dy=numder(y,h); dy0=dy(3);
coe=[];
for p =-2:2
a= (2*p^3-3*p^2-p+1)/12;     b= (4*p^3-3*p^2-8*p+4)/6;
c= (2*p^3-5*p)/2;
d= (4*p^3+3*p^2-8*p-4)/6;    e= (2*p^3+3*p^2-p-1)/12;
coe=[coe; [a -b c -d e]];
end;
[d1,d2]=size(y);
if ((min(d1,d2)>1) | (max(d1,d2) <5))
  error('Argument error in numder');
end;
dy =y;
dy(1)=(1/h)*sum(coe(1,:).*y(1:5));
dy(2)=(1/h)*sum(coe(2,:).*y(1:5));
for p=3:d2-2
   dy(p)=(1/h)*sum(coe(3,:).*y(p-2:p+2));
end;
dy(d2-1)=(1/h)*sum(coe(4,:).*y(d2-4:d2));
dy(d2)=(1/h)*sum(coe(5,:).*y(d2-4:d2));
% numder.m ends.

function ndf = numdf(fexpr,xval,h,order)
%NUMDF  gives numerical approximation of derivative
% of function given by fexpr. Order =1 or 2.
%       If the function is defined on (a,b) you must choose h
%       such that a < min(xval)-3*h and max(xval)+3*h < b.
% For order =1 the 5-point rule in Abramowitz-Stegun
```

```
% [AS], 25.3.6 is used.
% For order =2 the rule [AS], 25.3.24 is used.
% See also NUMDER.M
% USAGE: numdf('kk(x)',0.01:0.02:0.99, 1e-4)
%       or also: numdf('kk(x)',0.01:0.02:0.99, 1e-4,2)
% Argument name must be x.
%
coe =(1/12)*[ 1 -8 0 8 -1]; % coefficient for order =1
if (nargin ==4)
  if (order ==2)
    coe=(1/(12*h))*[ -1, 16, -30, 16, -1];
  end;                      % coefficient for order =2
end;
x1=xval;
x=[x1'-2*h,x1'-h,x1',x1'+h,x1'+2*h];
eval(['y='  fexpr  ';']);
dfc=(1/h)*(y*(coe'));
ndf=dfc';
%  numdf.m  ends.

function y = pic(fexpr)
% PIC plots the graphs of functions: MATLAB 5.3 (R11)
% USAGE: pic(fexpr) where fexpr is a MATLAB function expression
% Example: pic('sin(tan(x)) ')
% N.B. The argument name must be x.
% SAVES data in pic.dat
clf;
disp([' Enter now argument bounds x1, x2 for ' fexpr]);
t ='Please enter'; x1=input([ t ' x1: ']);
x2=input([ t ' x2: ']); x =x1: (x2-x1)/100: x2;
% N.B. x is a global variable name "sin(x)"
eval(['f = ' fexpr ';']);
axes('FontSize',[15],'FontWeight','bold'); hold on;
fig=plot(x,f); grid
title(['    y =  ' fexpr],'FontSize',[15],'FontWeight','bold');
xlabel('PIC.M','FontSize',[15],'FontWeight','bold');
delete pic.dat;            diary pic.dat;
disp(['f = ' fexpr]);
disp( ['     x       f     '] );
disp(['--------------------']);
disp([x' , f']);   diary off;
set(fig,'LineWidth',2.5);
y ='pic.m completed and data saved in pic.dat';
```



```
% pic.m  ends.

% FILE d116.m ends.
```

Output:

```
 Enter now argument bounds x1, x2 for cos(x)- numdf('sin(x)', x, 1e-4)
Please enter x1: -1
Please enter x2: 1
```