University of Turku / Department of Mathematics and Statistics

SCIENTIFIC COMPUTING

Exercise 06 / Solutions

1. The h-index is an index that attempts to measure both the productivity and impact of the published work of a scientist or scholar. The index is based on the set of the scientist's most cited papers and the number of citations that they have received in other publications (see http://en.wikipedia.org/wiki/H-index) The index is based on the distribution of citations received by a given researcher's publications. Hirsch writes:

"A scientist has index h if h of his/her N papers have at least h citations each, and the other $(N-h)$ papers have no more than h citations each."

(a) Suppose that Dr. Smart has 50 papers and paper $j, j = 1, ..., 50$ is cited c_j times. Suppose that $c_j = 1 + fix(60 * rand)$ for all $j, j = 1, ..., 50$. Find the h-index.

(b) Dr. Bright has written excellent 50 papers and the correponding citation counts are $c_j = j^p, j = 1, ..., 50$ for some $p > 0$. Find the corresponding index h_p and also plot h_p as a function of p when $p = 0.2*m, m = 1, ..., 20$.

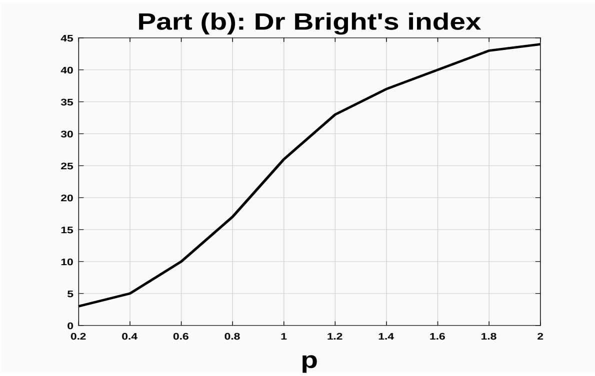
Solution:

```
% FILE: d061b.m begin
function w=d061b()
clear; close all
% Part (a) : Dr Smart
cc=[(1:50)' fix(1+99*rand(50,1))];
% Test data: With these citations, the index is 10:
% cc=[(1:50)' [zeros(40,1); 10*ones(10,1)]];
N=length(cc(:,1));
h = 0;
while (true)
  if sum(cc(:,2)>=h) <= h && sum(cc(:,2)<=h) >= N-h
    break;
  end
  h = h + 1;
end
fprintf('\nPart (a) Dr. Smart    Method I  Method 2\n')
fprintf('The index  is:          %6d   %6d  \n',h,hirsch(cc(:,2)))
fprintf('------------------\n')
fprintf('\nPart (b) Dr. Bright   Method I  Method 2\n')
% Part (b) Dr. Bright
hirscht=[];
for pp=1:10
p=0.2*pp;
cc=[(1:50)' ((1:50)').^p];
N=length(cc(:,1));
h = 0;
while (true)
  if sum(cc(:,2)>=h) <= h && sum(cc(:,2)<=h) >= N-h
    break;
  end
  h = h + 1;
end
fprintf(' P= %4.2f:               %6d   %6d  \n',p,h,hirsch(cc(:,2)))
hirscht=[hirscht  h];
end
figure
x=0.2*(1:10);
axes('FontSize',20, 'FontWeight','bold')

plot(x, hirscht,'LineWidth',2)
title(['Part (b): Dr Bright' char(39) 's index'],...
'FontSize',20, 'FontWeight','bold')
xlabel('p','FontSize',20, 'FontWeight','bold')
grid on
% Method 2 :
function h= hirsch(c)
m=length(c);
csort=sort(c,'descend');

n=1;
while ((csort(n) >n)&&(n<m))
      n=n+1;
end
h=n;
% FILE: d061b.m end
```

Output:

```
Part (a) Dr. Smart    Method I  Method 2
The index  is:             37       38
------------------
```

Part (b) Dr. Bright	Method I	Method 2
P= 0.20:	3	3
P= 0.40:	5	5
P= 0.60:	10	10
P= 0.80:	17	17
P= 1.00:	26	26
P= 1.20:	33	33
P= 1.40:	37	38
P= 1.60:	40	41
P= 1.80:	43	43
P= 2.00:	44	45



2. The daily temperature data is observed and the results appear in the table below. Create a file with these thirteen (x,y) pairs of this temperature data.

x	0.0	2.0	4.0	6.0	8.0	10.0	12.0	14.0	16.0	18.0	20.0	22.0	24.0
y	6.3	4.0	6.6	10.9	14.6	19.1	24.3	25.7	22.9	19.5	15.9	10.3	5.4

Let $m = \min\{y\}$, $M = \max\{y\}$ and set $a = 0.5*(M+m)$, $b = 0.5*(M-m)$. Try to find some reasonable integer value for the parameter c in the interval $[0, 24]$ so that the curve $y = a + b * \sin(2 * \pi * (x - c)/24))$ becomes as close to the data as possible. Carry out the following steps:

(a) Read the data (x_j, y_j), $j = 1, ..., 13$, from the file [or copy these values in a vector] and compute the maximum and minimum temperatures M and m. Then compute a and b.

(b) For each $c = 0 : 24$ compute

$$A(c) = \sum_{j=1}^{13} (y_j - (a + b * \sin(2 * \pi * (x_j - c)/24)))^2,$$

and choose the value of c that yields the minimal $A(c)$.

(c) With these values of the parameters a, b, c plot the curve $y = a + b * \sin(2 * \pi * (x - c)/24))$ and the data points in the same picture.

Solution:

```
% FILE d062.m begins.
clear
close all
% Read the temperatures from the file
load d032.dat;
x= d032(:,1);  m=length(x);
mytemp= d032(:,2);
for j=1:m
  fprintf('%5.1f',x(j))
end
fprintf('\n')
for j=1:m
  fprintf('%5.1f',mytemp(j))
end
fprintf('\n')
figure(1);
h=axes;
set(h,'FontWeight','bold','FontSize',20);
plot(x ,mytemp,'ko')
grid on
hold on
m=min(mytemp);
Ma=max(mytemp);
myA=[];
for c=0:24
  myt=mytemp-(0.5*(Ma+m)+0.5*(Ma-m)*sin(2*pi*(x-c)/24));
  myA=[myA norm(myt)^2];
end
c=0:24;
disp([c ; myA])
[minval,minind]=min(myA);
myc=c(minind);
disp(['c= ' num2str(myc) ', A(c)=' num2str(myA(minind))]);
myt=(0.5*(Ma+m)+0.5*(Ma-m)*sin(2*pi*(x-myc)/24));
txt=[num2str(0.5*(Ma+m)) '+' num2str(0.5*(Ma-m)) ...
     '*sin(2*pi*(x-' num2str(myc) ')/24))'];
plot(x,myt,'LineWidth',2);
xlabel(['c = ' num2str(myc) ],'FontSize',20,'FontWeight','bold' )
```

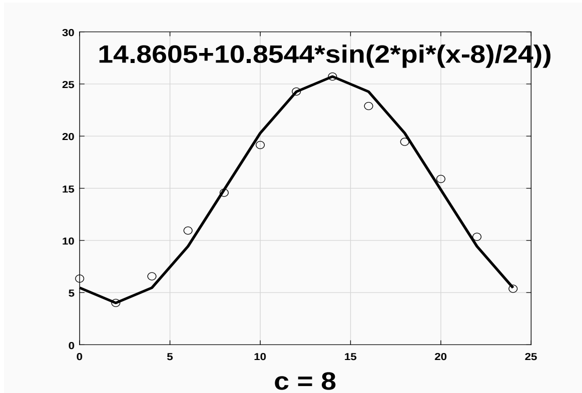
```
text(1, 28,txt,'FontSize',20,'FontWeight','bold')
print -dps d062.ps
parfmodel=@(t,lam)(lam(1)+lam(2)*sin(2*pi*(t-lam(3))/24));
y=mytemp;
laminit=[mean(y), 0.5*(max(y)-min(y)), 5];
parflam=myparf(x,y,parfmodel,laminit);
figure(2)
h=axes;
set(h,'FontWeight','bold','FontSize',20);
fobj=@(lam)norm(parfmodel(x,lam)-y);
disp(['fobj= ',num2str(fobj(parflam))])
%eval('mystr=char(92)')
disp(parflam')

plot(x,y,'k*',x, parfmodel(x,parflam),'k-','LineWidth',2)
title( ['lam= ' num2str(parflam)],'FontSize',20,'FontWeight','bold')
grid
print -dps d062b.ps
function w=myparf(xdata1,ydata1,mymodel, laminit)
% E.g. fmodel=@(x,lam)exp(-lam(1)* x)+lam(2)*exp(-lam(3)*x);
fobj=@(lam) norm(mymodel(xdata1,lam)-ydata1);

[lam fval eflag] = fminsearch(fobj,laminit);
w=[lam ];
end

% FILE d062.m ends.
```

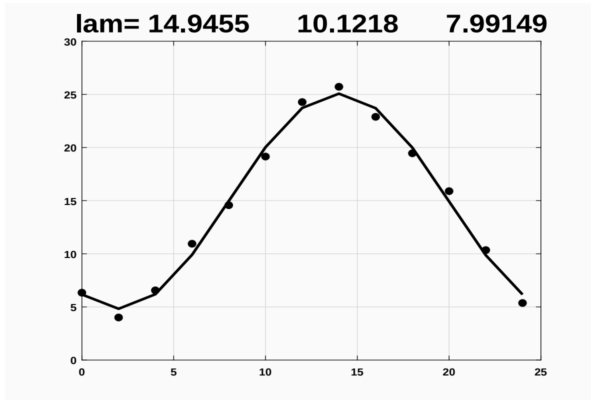
Output:

```
0.0  2.0  4.0  6.0  8.0 10.0 12.0 14.0 16.0 18.0 20.0 22.0 24.0
6.3  4.0  6.6 10.9 14.6 19.1 24.3 25.7 22.9 19.5 15.9 10.3  5.4
 1.0e+03 *

Columns 1 through 10

      0    0.0010    0.0020    0.0030    0.0040    0.0050    0.0060    0.0070    0.0080
  2.0602    1.7010    1.3358    0.9850    0.6668    0.3969    0.1895    0.0570    0.0101

Columns 11 through 20

  0.0100    0.0110    0.0120    0.0130    0.0140    0.0150    0.0160    0.0170    0.0180
  0.1984    0.4325    0.7469    1.1219    1.5303    1.9400    2.3171    2.6301    2.8533
```



```
 Columns 21 through 25

   0.0200    0.0210    0.0220    0.0230    0.0240
   2.9737    2.8686    2.6677    2.3905    2.0602

c= 8, A(c)=10.1386
fobj= 2.5153
   14.9455
   10.1218
    7.9915
```





Output:

3. To fit a circle (1) $(x-c_1)^2+(y-c_2)^2=r^2$ to n sample pairs of coordinates $(x_k, y_k), k=1,...,n$ we must determine the center (c_1, c_2) and the radius r. Now (1) $\Leftrightarrow$ (2) $2xc_1+2yc_2+(r^2-c_1^2-c_2^2)=x^2+y^2$. If we set $c_3=r^2-c_1^2-c_2^2$, then the equation takes the form

$$2xc_1+2yc_2+c_3=x^2+y^2 .$$

Substituting each data point we get

$$\begin{bmatrix} 2x_1 & 2y_1 & 1 \\ & \vdots & \\ 2x_n & 2y_n & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} x_1^2+y_1^2 \\ \vdots \\ x_n^2+y_n^2 \end{bmatrix}$$

This system can be solved in the LSQ-sense (if a is $m \times n, m > n$, then the LSQ solution of $a * x = b$ is given by

```
{ x = a \ b }
```

Then $r=\sqrt{c_3+c_1^2+c_2^2}$. Apply this algorithm for the points generated by

```
s=0.5+0.5*rand(10,1);
theta=2*pi*rand(10,1);
clear x
clear y
x=3*s.*cos(theta) ;
y=3*s.*sin(theta);
```

Plot the data and the fitted circle.

Solution:

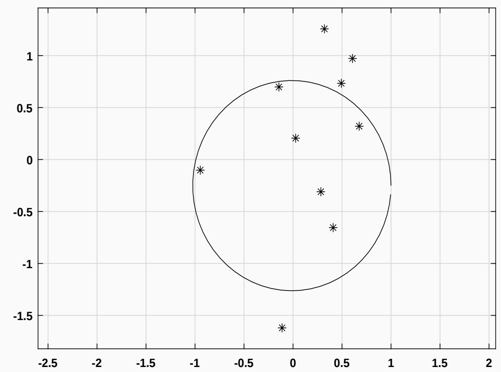
```
% FILE d063.m begins.
function h063
% USES: widemarg.m
close all
xy= randn(10,2);
fprintf('----------\nData 1:\n')
c=myfit(xy);
print -dps d063.ps
xy=xy+xy./abs(xy);
fprintf('----------\nData 2:\n')
c=myfit(xy);
print -dps d063b.ps
s=0.5+0.5*rand(10,1);
theta=2*pi*rand(10,1);
clear x
```



```
clear y
x=3*s.*cos(theta) ;
y=3*s.*sin(theta);
xy=[x y];
fprintf('----------\nData 3:\n')
c=myfit(xy);
print -dps d063c.ps
function c= myfit(xy)
figure
axes('FontSize',[18],'FontWeight','bold');
[d1,d2]=size(xy);
A=[xy  ones(d1,1)]; rhs=[xy(:,1).^2 + xy(:,2).^2];
c=A\rhs;
r=sqrt(c(3) + c(1)*c(1) + c(2)*c(2));
fprintf('center= (%8.4f,%8.4f), radius= %6.3f\n',c(1),c(2),r);
plot(xy(:,1), xy(:,2),'k*')
hold on
alf=0:0.1:6.2;
x=c(1)+r*cos(alf);
y=c(2)+r*sin(alf);
plot(x,y)
grid on
axis('equal')
widemarg(gcf)
% FILE d063.m ends.
```

Output:

```
----------
Data 1:
center= ( -0.0121, -0.2512), radius=  1.012
----------
Data 2:
center= ( -0.1867, -0.0558), radius=  2.294
----------
Data 3:
center= ( -0.4077,  0.2283), radius=  2.427
```



4. (a) The number of returned home works of the weekly problem sessions of the 1998 scientific computing course during the first six weeks were 21, 24, 17, 21, 14 and 17, respectively. Fit a model of the form

$$y = \lambda_1 \exp(-\lambda_2 x)$$

to this data and predict the number of participants in the 12th problem session.

(b) In 2018 in a similar course, during the first 5problem sessions these activity numbers were 23, 20, 19, 20, 16. Use this data to predict the number active students for week 6.

Solution:

```
% FILE  d064.m  begins.
% USES: d064mod, d064obj,
close all;clear;
global xdata;
global ydata;
xdata=1:6;
ydata=[21,24,17,21,14,17];
bar(xdata,ydata);
xscaled=(xdata)/12;
yscaled= ydata/25;
xdata=xscaled;
ydata=yscaled;
lam0=[3 1 ];           % Initial guess for lambda
y0=d064obj(lam0);      % Initial value of merit function
lam=fminsearch('d064obj',lam0);
```



```
                       % lam is the fitted value for
                       %   the parameter vector
disp([lam])
x=12*xdata;
yfit=25*d064model(lam, xscaled);
ysession12=25*d064model(lam, 1);
fprintf('Estimated number = %5d\n',fix(ysession12))
yfinal=d064obj(lam);  % Final value of the merit function
figure(1)
clf;
axes('FontSize',[20],'FontWeight','bold');

hold on;
title(['Object values: start = ' num2str(y0) ...
                    ', final = ' num2str(yfinal)],...
      'FontWeight','bold','FontSize',20)

bar(x,25*yscaled), colormap([127/255 1 212/255] );
plt=plot(x,yfit,'k-',7:12,25*d064model(lam,(7:12)/12),':k',...
         12,25*d064model(lam,12/12),'ko');
grid on;

txt1=' {\bf Fitted curve (solid)}';
text(x(2),yfit(2),txt1, 'FontWeight','bold','FontSize',20);

txt2=' {\bf Data points (bars)}';
text(x(4),yfit(4),txt2,'FontWeight','bold','FontSize',20);
% Unscale xdata ydata:
plot(12*xdata,25*ydata,'kd','MarkerSize',10);
ylabel('ydata: Number of students ','FontWeight','bold','FontSize',20);
xlabel('xdata: Number of week',...
       'FontWeight','bold','FontSize',20);
set(plt,'LineWidth',3);
disp(['Object values: start = ' num2str(y0) ...
                   ', final = ' num2str(yfinal)])
widemarg(gcf)
print -dps d064.ps

% Part (b): Use here parf11, parf18 style method
data2018=[23, 20, 19, 20, 16];
newmodel=@(t,lam)(lam(1)*exp(-lam(2)*t));
weeks=[1  2   3  4   5];
lam0=[18,1];
```

```
lam=myparf(weeks, data2018,newmodel,lam0)
figure(2)
%axes('FontSize',[20],'FontWeight','bold');
axes('FontSize',[20],'FontWeight','bold');
bar(weeks,data2018)
hold on
title(['lam= ' sprintf('  %6.4f',lam) '(using parf11 style)'],...
    'FontSize',[20],'FontWeight','bold')
plot([weeks 6], newmodel([weeks 6], lam),'k-','LineWidth',2)
xlabel(['Demo 6: ' num2str(fix(newmodel(6,lam))) ' = number of passing students?'],...
    'FontSize',[20],'FontWeight','bold')
grid
print -dps d064b.ps
function w=myparf(xdata1,ydata1,mymodel, laminit)
% E.g. fmodel=@(x,lam)exp(-lam(1)* x)+lam(2)*exp(-lam(3)*x);
fobj=@(lam) norm(mymodel(xdata1,lam)-ydata1);

[lam fval eflag] = fminsearch(fobj,laminit);
w=[lam ];
end
% FILE  d064.m  ends.

function y=d064model(lam,x)
y = lam(1)*exp(-lam(2)*(x));
% FILE  d064model.m  ends.

function y=d064obj(lambda)
% USES: d064model
global xdata;
global ydata;
y = norm(d064model(lambda,xdata) - ydata);
% FILE  d064obj.m  ends.

% FILE  d064inline.m  begins.
clear
%path(path,'../../util')
fmodel=inline('lam(1)*exp(-lam(2)*x)','x','lam');
fobj=inline('norm(feval(fmodel,x,lam)-y)','lam','fmodel','x','y');

xdata=1:6;            ydata=[21,24,17,21,14,17];
bar(xdata,ydata);
xscaled=(xdata)/12;   yscaled= ydata/25;   % SCALE DATA
xdata=xscaled;        ydata=yscaled;
lam0=[3 1 ];           % Initial guess for lambda
```

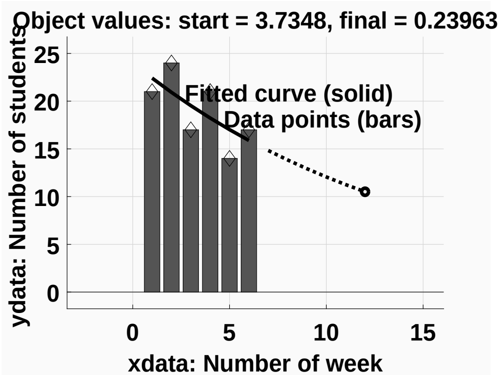


```
y0=fobj(lam0,fmodel,xdata,ydata);  % Initial value of fobj

lam  = fminsearch(fobj,lam0,[],fmodel,xdata,ydata);

                       % lam is the fitted value for
                       %   the parameter vector
fprintf('lam =');     fprintf(' %6.3f',lam);   fprintf('\n')
x=12*xdata;                    % Unscale data
yfit=25*fmodel( xscaled,lam);  % Unscaled fitted values
ysession12=25*fmodel(1,lam);   % Estimated value for session 12
fprintf('Estimated number = %5d\n',fix(ysession12))
yfinal=fobj(lam,fmodel,xdata,ydata);  % Final value of fobj
figure(1)
clf;
axes('FontSize',[20],'FontWeight','bold');     hold on;
title(['Object values: start = ' num2str(y0) ...
                    ', final = ' num2str(yfinal)],...
      'FontWeight','bold','FontSize',20)

bar(x,25*yscaled), colormap([127/255 1 212/255] );
plt=plot(x,yfit,'k-',7:12,25*fmodel((7:12)/12,lam),':k',...
         12,25*fmodel(12/12, lam),'ko');  grid on;

txt1=' {\bf Fitted curve (solid)}';
text(x(2),yfit(2),txt1, 'FontWeight','bold','FontSize',20);

txt2=' {\bf Data points (bars)}';
text(x(4),yfit(4),txt2,'FontWeight','bold','FontSize',20);
% Plot unscaled xdata ydata:
plot(12*xdata,25*ydata,'kd','MarkerSize',10);
ylabel('ydata: Number of students ','FontWeight','bold','FontSize',20);
xlabel('xdata: Number of week (D064inline, MME05)',...
       'FontWeight','bold','FontSize',20);
set(plt,'LineWidth',3);
disp(['Object values: start = ' num2str(y0) ...
                   ', final = ' num2str(yfinal)])
widemarg(gcf)
pause(3)
% FILE  d064inline.m  ends.
```

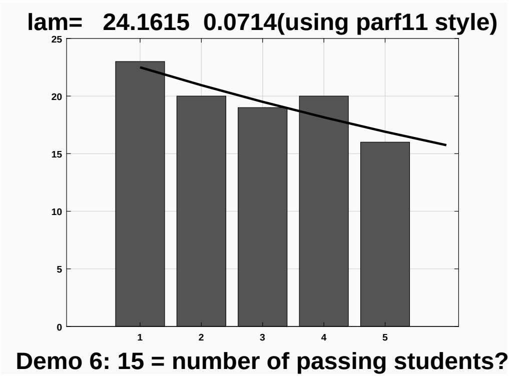
Output:

```
 0.9606    0.8265
```

```
Estimated number =    10
Object values: start = 3.7348, final = 0.23963

lam =

   24.1615    0.0714
```



Using parfit style:



5. Recall from linear algebra that if $\Delta \equiv ad - bc \neq 0$, then

$$\Delta \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

Consider the Newton method $x_{n+1} = x_n - J_f(x_n)^{-1} f(x_n)$ to solve the system

$$\begin{cases} x_1^2 + x_2^2 - 1 &= 0 \\ x_1^2 - x_2^2 &= 0 \end{cases}$$

with the initial point $(0.5, 0.5)$. Use standard methods from linear algebra and multidimensional calculus, such as the above formula for matrix inverse, to compute the following expressions analytically and to print the numerical values $J_f(x_n), J_f(x_n)^{-1}, T(x_n) \equiv J_f(x_n)^{-1} f(x_n)$ for the first three iteration steps.

Solution:

```
function z=d065new(x, nriter)
  for j=1:nriter
     disp('J_f(x) =')
     a=J(x); b=invJ(x); c= b*F(x);
      disp(a)
     disp('J_f(x)^{-1} =')
     disp(b)
     disp('T(x) = J_f(x)^{-1} *F(x) =')
     disp(c)
     disp('J(x)*invJ(x) =')
     disp(a*b)
     disp('J_f(x)^{-1} *F(x)-T(x) =')
     disp(c-T(x))
     x= x-c;
     disp('x =')
     disp(x)
     pause(1)
  end

function w= F(x)
  w=[x(1)*x(1)+ x(2)*x(2)-1; x(1)*x(1)- x(2)*x(2)];

function w= J(x)
  w=2*[x(1)  x(2); x(1) -x(2)];

function w=invJ(x)
  w=-(1/(4*x(1)*x(2)))*[-x(2) -x(2); -x(1) x(1)];

function w=T(x)   % T= invJ(x)*F(x)
  w=[ (-1+2*x(1)*x(1))/(4*x(1)) ; (-1+2*x(2)*x(2))/(4*x(2))];
```

Output:

```
J_f(x) =
     1     1
     1    -1

J_f(x)^{-1} =
    0.5000    0.5000
    0.5000   -0.5000

T(x) = J_f(x)^{-1} *F(x) =
   -0.2500
   -0.2500

J(x)*invJ(x) =
     1     0
     0     1

J_f(x)^{-1} *F(x)-T(x) =
     0
     0

x =
    0.7500
    0.7500

J_f(x) =
    1.5000    1.5000
    1.5000   -1.5000

J_f(x)^{-1} =
    0.3333    0.3333
    0.3333   -0.3333

T(x) = J_f(x)^{-1} *F(x) =
    0.0417
    0.0417

J(x)*invJ(x) =
     1     0
     0     1

J_f(x)^{-1} *F(x)-T(x) =
```



```
     0
     0

x =
    0.7083
    0.7083

J_f(x) =
    1.4167    1.4167
    1.4167   -1.4167

J_f(x)^{-1} =
    0.3529    0.3529
    0.3529   -0.3529

T(x) = J_f(x)^{-1} *F(x) =
    0.0012
    0.0012

J(x)*invJ(x) =
    1.0000         0
         0    1.0000

J_f(x)^{-1} *F(x)-T(x) =
   1.0e-18 *

    0.2168
    0.2168

x =
    0.7071
    0.7071
```

6. The program hlp066new.m on the www-page plots a curve through given points. Use it to plot the shape of your hand using sufficiently many points, e.g.25-30 points. Experiment with the program by changing pchip to spline and other methods of interpolation.

Solution:

```
% FILE d066new.m begins.
clear
close all
x=[2.6 1.6 1.0 0.5 0.6 1.1 2.4 2.5 2.7 3.0 3.2  3.4 3.7 ...
3.9 4.0 4.2 4.4 4.6 5.0 6.0 6.2 6.3 5.6 5.2 4.0 2.6];
y=[0.5 2.1 3.1 4.2 4.5 4.4 3.0 7.0 7.2 7.0 4.5 7.4 7.6 ...
7.4 4.5 7.1 7.2 7.1 4.0 5.8 5.9 5.6 3.1 0.5 0.2 0.5];
m=length(x);
xi =1:0.05:m;
pi=interp1(1:m,x,xi,'pchip');
qi=interp1(1:m,y,xi,'pchip');
subplot(1,2,1)
plot(x,y,'r.',pi,qi,'b-')
title(['pchip'])
axis('equal')
xi =1:0.05:m;
pi=interp1(1:m,x,xi,'spline');
qi=interp1(1:m,y,xi,'spline');
subplot(1,2,2)
plot(x,y,'r.',pi,qi,'b-')
title('spline')
axis('equal')
% FILE d066new.m ends.
```

Output: