University of Turku / Department of Mathematics and Statistics
SCIENTIFIC COMPUTING
Exercise 05 / Solutions

1. (a) The program hlp051.m produces a picture of a well-known distribution. What is this distribution?

(b) The function $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-y^2) dy$ is a built-in function of MATLAB. Show by MATLAB experiments that for $a, b \in R, a \neq 0, x_1 < x_2$,

$$\int_{x_1}^{x_2} \exp\left(-\frac{(x-b)^2}{(2a^2)}\right) dx = a\sqrt{\frac{\pi}{2}}\left(erf(\frac{x_2-b}{a\sqrt{2}}) - erf(\frac{x_1-b}{a\sqrt{2}})\right) .$$

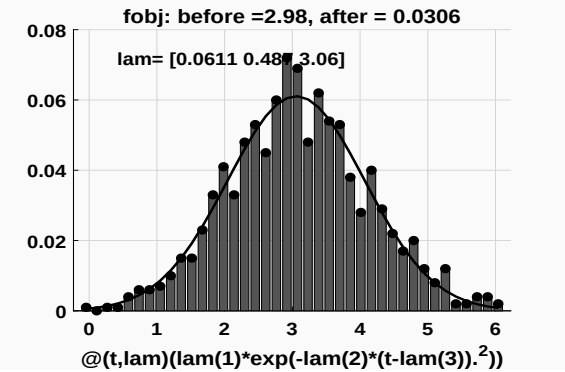
This can also be verified by change of variable.

Solution:

```
% FILE  hlp051.m  begins.
clear; close all;
N=1000;
u=rand(1,N); v=rand(1,N);
z1=sqrt(-2*log(u)).*cos(2*pi*v);
z2=sqrt(-2*log(u)).*sin(2*pi*v);
data=z1+3;
a=histogram(data,40); % 40 is number of bins
hlp=a.BinEdges;
y=a.Values;    % sum(y)=1000; length(hlp) =1+length(y)
y=y/sum(y);    %sum(y) = 1
x=hlp(2:end);  % length(x) =length(y)
fmodel=@(t,lam) (lam(1)*exp(-lam(2)* (t-lam(3)).^2));
fobj=@(lam) norm(fmodel(x,lam)-y);
lam0=[-1,1,2];
v1=fobj(lam0);  % Value before fitting
lam=myparf(x, y,fmodel,lam0)
v2=fobj(lam);   % Value after fitting
figure
fprintf('fobj  before/after fitting: %f   %f\n',v1,v2);
yfit=fmodel(x,lam);
axes('FontSize',[15],'FontWeight','bold'); hold on;
bar(x,y)
hold on
title(['fobj: before =' num2str(v1,3) ', after = ' ...
       num2str(v2,3)])
plot(x,yfit,'k-',x,y,'k.','MarkerSize',20,'LineWidth',2);
disp(['fmodel >data: ' num2str(sum(yfit>y)) '; fmodel <= data: ' num2str(sum(yfit<=
grid on;
ax=axis; x1=ax(1)+0.1*(ax(2)-ax(1)); y1=ax(3)+0.9*(ax(4)-ax(3));
text(x1,y1, ['lam= ' mat2str(lam,3)],'FontWeight','bold','FontSize',[15]);
xlabel(func2str(fmodel))
%lam(2) =1/(2*sig^2)
sighat=1/sqrt(2*lam(2));
fprintf('muhat, sighat = %f   %f\n', lam(3),sighat)



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  parfhisto.m  ends.

% FILE d051new.m begins.
function y=d051new
clear
myf2=inline('exp(-(x-b).^2/(2*a*a))','x','a','b');
a= 10*rand;
b=5*(rand-0.5);
fprintf(' %8s  %12s  %12s  %12s  %12s \n', 'x1', 'x2', 'a', 'b','LHS-RHS')
d051f1=@(x) (1/(sqrt(pi))*(exp(-x.*x ))-erf(x));
d051f2= @(x)(exp(-(x-b).^2/(2*a*a)));
for j=1: 10
  x1= 3*rand;
  x2= x1+10*rand;
  LHS=quadl(d051f2,x1, x2, 1e-10);
  LHS2=quadl(myf2,x1, x2, 1e-10,[],a,b);  % Now LHS-LHS2 = 0
  RHS=erf((x2-b)/(a*sqrt(2))) - erf((x1-b)/(a*sqrt(2)));
  RHS=RHS*a*sqrt(pi/2);
  fprintf(' %8.6f  %12.6f  %12.6f  %12.6f  % 12.4e\n', ...
      x1, x2, a, b, LHS-RHS)
end


% FILE d051new.m ends.
```

Output:

```
lam =

    0.0611    0.4871    3.0567

fobj  before/after fitting: 2.977460   0.030555
fmodel >data: 19; fmodel <= data: 21
muhat, sighat = 3.056686   1.013153
       x1            x2             a             b       LHS-RHS
 0.756030     10.019176      4.314272     -0.296189    5.8620e-14
 2.655769      4.665096      4.314272     -0.296189    6.0174e-14
 1.944588      8.619022      4.314272     -0.296189    3.1086e-15
 0.421687      9.206014      4.314272     -0.296189    5.1514e-14
 1.388837      3.359691      4.314272     -0.296189    2.2649e-14
 0.090558      3.232163      4.314272     -0.296189   -1.0010e-12
 0.706699      1.891420      4.314272     -0.296189    4.4409e-16
 2.031913      2.506399      4.314272     -0.296189    2.7756e-16
 1.110719     10.940014      4.314272     -0.296189    4.2633e-14
 2.527954     11.564694      4.314272     -0.296189   -2.5757e-14
```



2. On the www-page is given the program `hpl052.m` which compares two methods of numerical integration, namely Riemann's sum and Simpson's Rule, over a rectangular region in the plane with the test function $f(x, y) = xy$. The program prints the error.

(a) Modify the program to use the function $g(x, y) = \sin(2x) * \cos(4 * y)$ and report the results.

(b) Write the code also for the Trapezoid Rule and one of MATLAB's built-in function `dblquad, quadl, integral,...` and report the error.

Provide an order or preference of the methods based on the accuracy of each method.

Solution:

```
% FILE d052.m begins.
function h052
a=0; b=4; c=0;d=1;   % bounds of integration
exact=(1/8)*(cos(2*a)- cos(2*b))*(sin(4*d)-sin(4*c));
                     % exact result
res=dblquad('sin(2*x).*cos(4*y)',a,b,c,d);
fprintf('Error using dblquad: %12.3e\n', exact-res);

fprintf('Error using doubleint0 (Riemann sum):\n');
fprintf('n          20           60           100  \n')
for m=20:40:100
fprintf('%3d  ',m)
for n=20:40:100
   numer=doubleint0(a,b,c,d,m,n);
   fprintf('  %12.3e', numer-exact);
end
fprintf('\n')
end



fprintf('Error using doubleint2 (Simpson Rule):\n')
fprintf('n          20           60           100  \n')
for m=20:40:100
fprintf('%3d  ',m)
for n=20:40:100
         numer=doubleint2(a,b,c,d,m,n);
         fprintf('  %12.3e', numer-exact);
end
fprintf('\n')
end
fprintf('Error using doubleint3 (Trapezoid Rule):\n')
fprintf('n          20           60           100  \n')
for m=20:40:100
fprintf('%3d  ',m)
for n=20:40:100
         numer=doubleint3(a,b,c,d,m,n);
         fprintf('  %12.3e', numer-exact);
end
fprintf('\n')
```

```
end

function z= myf(x,y)
z=sin(2*x).*cos(4*y);

function val=doubleint0(x1,x2,y1,y2,m,n)
% Riemann Sum
a=x1; b=x2;
s=0.0;
dx=(b-a)/m;
dy=(y2-y1)/n;
for ii=0:(m-1)
      x=a+ii*dx;
for jj=0:(n-1)
      y=y1+jj*dy;
      s=s+myf(x,y);
end
end
val=s*dx*dy;

function w=doubleint2(x1,x2,y1,y2,m,n)
% Note calling program must supply myf
% Compute \int_x1^x2  \int_{y1}^{y2} myf(x,y) dx dy
% This is based on the one-dimensional Simpson rule
% \int_a^b f(x) dx = (h/3) [y_0+ 4y_1 + 2y_2+ 4y_3 + ...+2y_{2n-2}+
%                       4 y_{2n-1}+ y_{2n}]
% x_k= a+ k*(b-a)/(2n),   y_k=f(x_k), k=0,..., 2n
a=x1; b=x2;
s=0.0;
wx=ones(1,1+m);
wx(2:2:(end-1))=4*ones(size(wx(2:2:(end-1))));
wx(3:2:(end-2))=2*ones(size(wx(3:2:(end-2))));
wx=(1/3)*wx;
wy=ones(1,1+n);
wy(2:2:(end-1))=4*ones(size(wy(2:2:(end-1))));
wy(3:2:(end-2))=2*ones(size(wy(3:2:(end-2))));
wy=(1/3)*wy;

for ii=0:(m)
      x=a+ii*(b-a)/m;
      dy=(y2-y1)/n;
for jj=0:(n)
      y=y1+jj*dy;
```



```
      s=s+wx(ii+1)*wy(jj+1)*myf(x,y)*dy*(b-a)/m;
end
end
w= s;

function w=doubleint3(x1,x2,y1,y2,m,n)
% Katso
dx=(x2-x1)/m;  x=x1:dx:x2;
dy=(y2-y1)/n;  y=y1:dy:y2;
s=0.0;
sy=zeros(1,m+1);
for ii=1: (m+1)
      xx=x(ii);
      z=myf(xx,y);
      sy(ii)=0.5*(sum(z(2:end).*diff(y)) + sum(z(1:(end-1)).*diff(y)));
end
w= 0.5*(sum(sy(2:end).*diff(x))+ sum(sy(1:(end-1)).*diff(x)));
% FILE d052.m ends.
```

Output:

```
Error using dblquad:   -1.344e-09
Error using doubleint0 (Riemann sum):
n          20           60          100
 20       3.973e-02     2.662e-02     2.403e-02
 60       2.902e-02     1.386e-02     1.087e-02
100       2.700e-02     1.146e-02     8.385e-03
Error using doubleint2 (Simpson Rule):
n          20           60          100
 20      -1.668e-05    -1.572e-05    -1.571e-05
 60      -1.159e-06    -2.026e-07    -1.922e-07
100      -9.925e-07    -3.658e-08    -2.622e-08
Error using doubleint3 (Trapezoid Rule):
n          20           60          100
 20       1.805e-03     1.488e-03     1.463e-03
 60       5.215e-04     2.007e-04     1.750e-04
100       4.191e-04     9.792e-05     7.224e-05
```

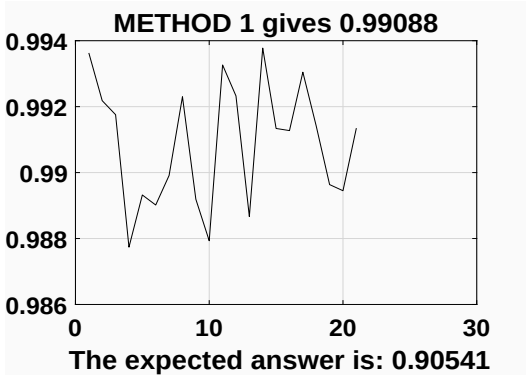
3. Generate N pairs of points A, B in the unit disk $\{(x, y) : x^2 + y^2 < 1\}$ and print the mean value of the distances. Repeat this for $N = 1000 : 1000 : 30000$.

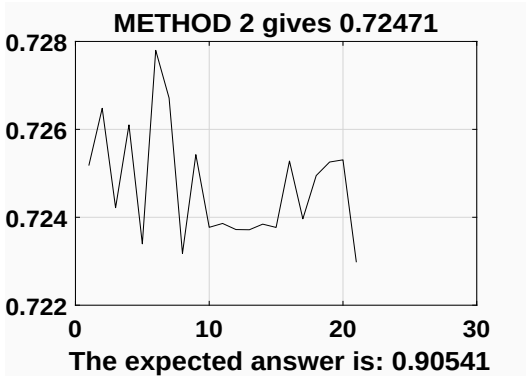
Solution:

```
%FILE: d053new.m
```

```
clear; close all;

for ii=1:2
    figure
    v=[];
for N=50000:1000: 70000
if ii==1
    %METHOD 1:
x=-1+2*rand(1,N); y=-1+2*rand(1,N).*sqrt(1-x.^2);
A=x+i*y;
x=-1+2*rand(1,N); y=-1+2*rand(1,N).*sqrt(1-x.^2);

B=x+i*y;
else
%METHOD 2:
A=rand(1,N).*exp(i*rand(1,N)*2*pi);
B=rand(1,N).*exp(i*rand(1,N)*2*pi);
end
dist=abs(A-B);
v=[v;  mean(dist)];
end

%v
plot(v,'k-')
grid
title(['METHOD ' num2str(ii) ' gives ' num2str(mean(v))])
disp(['Grimmett-Stirzaker: p. 139:  128/(45 pi) ' num2str(128.0/(45*pi))])
xlabel([' The expected answer is: ' num2str(128/(45*pi))])
if ii==1
print -dps d0531.ps
else
    print -dps d0532.ps
end

end

%FILE: d053new.m
```

Output:





4. Suppose that A is a non-singular $n \times n$ matrix with columns $A^{(j)}, j = 1, ..., n$, and x and b are $n \times 1$ vectors. By Cramer's Rule, the solution to $Ax = b$ is given by

$$x_j = \left((\det(A))^{-1} \right) \det(C_j) \, , C_j = [A^{(1)} A^{(2)} ... A^{(j-1)} b A^{(j+1)} ... A^{(n)}] \, .$$

Verify this procedure with MATLAB tests for small n. For how big values of n this is a reasonable procedure?

Solution:

```
% FILE d054.m begins.
% Verify Cramer's Rule
for k=1:2
t0=clock;
```

```
for n=10:10:100
a=rand(n,n); exa=rand(n,1); b=a*exa;
if (k==1)
mydet=det(a);
xcra=[];
for j=1:n
    cj=[a(:, 1:(j-1)) b a(:, (j+1):n)];
    xcra=[xcra; det(cj)/mydet];
end
end
sol=a\b;
if (k==1)
fprintf('%3d. Error = %12.4e\n',n,norm(xcra-sol))
else
fprintf('%3d. Error = %12.4e\n',n,norm(exa-sol))
end
end
if (k==1)
fprintf('Using Cramers method, time =%5.3f\n', etime(clock,t0) ) ;
else
fprintf('Using MATLAB, time =%5.3f\n', etime(clock,t0) ) ;
end
end
% FILE d054.m ends.
```

Output:

```
 10. Error =   4.4533e-15
 20. Error =   4.7244e-13
 30. Error =   2.1565e-14
 40. Error =   1.2132e-13
 50. Error =   5.3188e-13
 60. Error =   5.1097e-14
 70. Error =   1.5124e-13
 80. Error =   1.7206e-13
 90. Error =   2.8458e-13
100. Error =   3.7245e-13
Using Cramers method, time =0.298
 10. Error =   2.5335e-15
 20. Error =   1.9281e-14
 30. Error =   1.2281e-14
 40. Error =   1.0587e-13
 50. Error =   5.8315e-14
 60. Error =   1.2235e-13
```



```
 70. Error =   3.1610e-13
 80. Error =   3.3174e-13
 90. Error =   4.6384e-13
100. Error =   2.4612e-13
Using MATLAB, time =0.010
```

5. The capacity of an oil container is 3000 l and its shape is a right circular cylinder. The cross section is a circle with diameter 1.3 m. Find the height of the container. The symmetry axis of the container is horizontal. Write a table of the form

```
    n      Measurement/m   Amount of oil/1000 l
-----------------------------------------------
    1      0               0
    2      0.05            0.0379
 ......

   14      0.65            1.5
```

Here the measurement column indicates how much oil is left in the container in meters and the next column indicates the volume of the oil. What is the measurement when 1000 l oil is remaining?

Solution:

```
% FILE d055b.m
% The area of the set
% x^2+y^2 <1  && x > 1-s  (0<s<1) is
% A(s) = pi(2*a /(2*pi)  -(1-s)sqrt(1-(1-s)^2))=
% acos(1-s)-(1-s).*sqrt(2*s-s.*s )
% Hence the area of x^2+y^2 <(d/2)  && x > (d/2)-t  (0<t<(d/2)) is
% A(t/(d/2))*(d/2)^2
function w=oil()
V=3; d= 1.3;
for kk=1:14
    t=0.05*(kk-1);
    fprintf('%3d  %6.4f   %6.4f\n',kk,t, myoil(V,d,t))
end


myf=@(t)(myoil(V,d,t)-1);
rt=fzero(myf,0.5);
```

```
fprintf('When t= %6.3f m, 1000 l is left in the container\n',rt);



function my=myoil(v,d,t)

myxsection=@(s)(acos(1-s)-(1-s).*sqrt(2*s-s.*s ));

h=3/(pi*(d/2)^2);
if t<=d/2
  my=h*(myxsection(t/(d/2)))*(d/2)^2;
else
    my=v-h*myxsection((d/2-t)/(d/2))*(d/2)^2;
end
% FILE d055b.m
```

Output:

```
  1  0.0000   0.0000
  2  0.0500   0.0380
  3  0.1000   0.1061
  4  0.1500   0.1926
  5  0.2000   0.2927
  6  0.2500   0.4038
  7  0.3000   0.5237
  8  0.3500   0.6509
  9  0.4000   0.7840
 10  0.4500   0.9218
 11  0.5000   1.0632
 12  0.5500   1.2073
 13  0.6000   1.3532
 14  0.6500   1.5000
When t=  0.478 m, 1000 l is left in the container
```

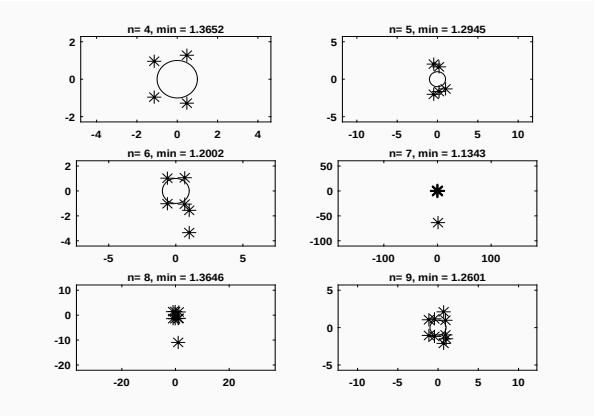
6. Polynomial Inequalities Springer-Verlag, 1995 states that if $p(z) = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_0$ and $a_0 \geq a_1 \geq \ldots \geq a_n > 0$ then all zeros of p lies outside the open unit disk. Verify experimentally this statement by generating random coefficients a_j and by plotting the roots in the plane. Polynomials of this type occur e.g. in the analysis of time series.

Solution:


```
% FILE d056.m begins.
close all

set(0,'DefaultAxesFontWeight','bold')
set(0,'DefaultAxesFontSize',[8])
set(0,'DefaultTextFontSize',[8])

the=0:0.1:6.3;
x=cos(the);  y=sin(the);
for n=4:9
a0=5*rand(1,n+1);
a=sort(a0);
z=roots(a);
subplot(3, 2, n-3);
plot(x,y,'b-')
mi= min(abs(z));
title(['n= ' num2str(n) ', min = ' num2str(mi) ],...
      'FontSize',[8],'FontWeight','bold')
fprintf('min(abs(root))= %12.5f\n',mi);
hold on
for j=1:n
plot(z(j),'k*','MarkerSize', 10)
end
widemarg(gcf)
axis equal
end
% FILE d056.m ends.
```

Output:

```
min(abs(root))=      1.36519
min(abs(root))=      1.29453
min(abs(root))=      1.20015
min(abs(root))=      1.13429
min(abs(root))=      1.36459
min(abs(root))=      1.26008
```


n= 4, min = 1.3652
n= 5, min = 1.2945
n= 6, min = 1.2002
n= 7, min = 1.1343
n= 8, min = 1.3646
n= 9, min = 1.2601