University of Helsinki / Department of Mathematics and Statistics

SCIENTIFIC COMPUTING

Exercise 01 / Solutions

1. Apply the recursion formula $x_0 = 1, x_{n+1} = \frac{1}{2}(x_n + \frac{a}{x_n}), n = 0, 1, 2, \ldots$ for $\sqrt{a}$ to compute $\sqrt{3}$. Print the results in the following format:

```
n  x(n)   Error
0  1
.....
6  ...
```

Solution:

```
% FILE d011 .m begins.
x=1; a=3;
fprintf('    n     x(n)            Error\n')
for j=1:6
%    disp([j-1, x, x-sqrt(3)]);
    fprintf('   %2d%16.10f   %12.4e\n',j-1,x,x-sqrt(3));
    x=0.5*(x+a/x);
end
% FILE d011.m ends.
```

Output:

```
    n     x(n)             Error
    0    1.0000000000    -7.3205e-01
    1    2.0000000000     2.6795e-01
    2    1.7500000000     1.7949e-02
    3    1.7321428571     9.2050e-05
    4    1.7320508100     2.4459e-09
    5    1.7320508076     0.0000e+00
```

2. Approximations to the number π are given by the formula

$$p(n) = \sum_{k=0}^{n} \frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right).$$

Print the first few results in the same format as in problem 1.

Solution:

```
% FILE d012.m begins.
s=0;
fprintf('%3s  %12s  %12s\n', 'n', 'p(n)', 'Error')
for k=0:10
    fprintf('%3d  %12.8f  %12.4e\n',k,s,s-pi);
    s=s+(16^(-k))*(4/(8*k+1)- 2/(8*k+4)-1/(8*k+5)-1/(8*k+6))
end
% FILE d012.m ends.
```

Output:

```
  n          p(n)         Error
  0     0.00000000   -3.1416e+00
  1     3.13333333   -8.2593e-03
  2     3.14142247   -1.7019e-04
  3     3.14158739   -5.2632e-06
  4     3.14159246   -1.9602e-07
  5     3.14159265   -8.1295e-09
  6     3.14159265   -3.6171e-10
  7     3.14159265   -1.6912e-11
  8     3.14159265   -8.2023e-13
  9     3.14159265   -4.0856e-14
 10     3.14159265   -1.7764e-15
```

3. In Solmu 2/2005 (http://solmu.math.helsinki.fi/2005/2/) following problem was studied. Is it true that a continuous function $(0, \infty) \to (0, \infty)$ satisfying the conditions:

1. $f(2x) = 2f(x)$, and
2. $f(1) = c$

is always of the form $f(x) = cx$. In the article, the following counterexam was presented:

$$f(x) = 2^{-n}x^2 + 2^{n+1} \text{ for } x \in [2^n, 2^{n+1}),$$

where $n = 0, \pm 1, \pm 2, \ldots$. Plot the graph of this function.

Solution:

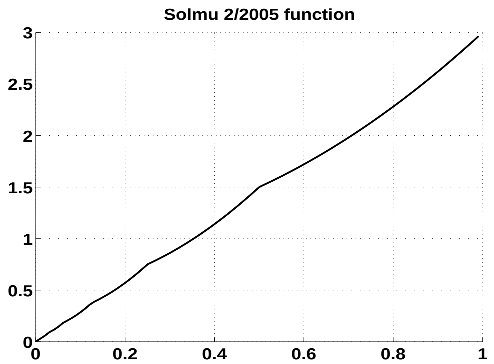
```
% FILE d013.m begins.
% MME05 harj 1 teht 3
```

```
% USES: h013f.m
close all
m=10;
x= (2^(-m)):0.01:1;
y=h013f(x);
figure
axes('FontSize',[18],'FontWeight','bold');
hold on
plot(x,y,'k-','LineWidth',2)
grid on
title('Solmu 2/2005 function','FontSize', 18,'FontWeight','bold');
% FILE d013.m ends.

function y=h013f(x)
     n=fix(floor(log2(x)));
     y=2.^(-n).*(x.^2)+ 2.^(n+1);


% FILE h013f.m ends.
```

Output:



4. Let $(x_j, y_j), j = 0, 1, ..., n$ be the vertices of a polygon with $(x_0, y_0) = (x_n, y_n)$. The area of the polygon is given by $a = \frac{1}{2}\sum_{i=1}^{n} t_i$ with $t_i = x_{i-1}y_i - x_i y_{i-1}$. Carry out the following steps for each of the regular polygons triangle, square and hexagon:

(a) Choose vertices and compute the area by school geometry.
(b) Compute the area by the formula and compare to the exact value.

(c) Plot the figure.

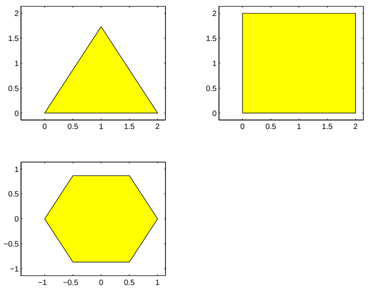
Solution:

```
% FILE d014.m begins.
% USES: widemarg.m
path(path,'../../util')
close all
disp('Triangle')
x=[0, 2, 1, 0]; y=[0, 0, sqrt(3), 0];
fprintf('d014area = %12.5e  , exact area = %12.5e \n', ...
d014area(x,y),sqrt(3))
%figure
subplot(2,2,1)
plot(x,y)
widemarg(gcf)
patch(x,y,'y')
disp('Square')
x=[0, 2, 2, 0, 0]; y=[0, 0, 2, 2, 0];
fprintf('d014area = %12.5e  , exact area = %12.5e \n', ...
d014area(x,y),4)
%figure
subplot(2,2,2)
plot(x,y)
widemarg(gcf)
patch(x,y,'y')
disp('Hexagon')
k=0:6;
x=real(exp(i*pi*k/3)); y=imag(exp(i*pi*k/3));
fprintf('d014area = %12.5e  , exact area = %12.5e \n', ...
d014area(x,y),1.5*sqrt(3))
%figure
subplot(2,2,3)
plot(x,y)
widemarg(gcf)
patch(x,y,'y')
print -dps d014.ps
% FILE d014.m ends.
```

Output:

```
Warning: Name is nonexistent or not a directory: ../../util.
```

```
> In /usr/local/matlab/toolbox/matlab/general/path.m at line 116
  In /home/vuorinen/mme08/demo08/d01/d014.m at line 3
  In /home/vuorinen/mme08/demo08/d01/d01all.m at line 28
Triangle
d014area =  1.73205e+00  , exact area =  1.73205e+00
Square
d014area =  4.00000e+00  , exact area =  4.00000e+00
Hexagon
d014area =  2.59808e+00  , exact area =  2.59808e+00
```



5. Hilbert's inequality says that for $a_k, b_k \geq 0$

$$\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\frac{a_m b_n}{m+n+1} \leq \pi(\sum_{m=0}^{\infty} a_m^2)^{1/2}(\sum_{n=0}^{\infty} b_n^2)^{1/2}.$$

Carry out a numerical verification of this inequality.

Solution:

```
% FILE d015.m begins.
% MME05 harj 1 teht 5
% Use random numbers such that the RHS series converge
% Choose mmax terms for each sum, mmax large
% Use terms a_k = rand *1/k, b_k = rand *1/k
close all
for tst=1: 10     % counter for test number
mmax=1000+fix(10000*rand);
a= (1:mmax); a=(1./a).*rand(1,mmax);
b= (1:mmax); b=(1./b).*rand(1,mmax);
rhs1=sum(a.^2); rhs2=sum(b.^2);
lhs=0;           %The values of the left hand side
                 % will be accumulated in this variable
```



```
for ii=1:mmax
for jj=1:mmax
lhs= a(ii)*b(jj)/((ii-1)+(jj-1) +1);
end              % end of jj loop
end              % end of ii loop
rhs=pi*sqrt(rhs1*rhs2); % The right hand side of the inequali
fprintf('%2d. Test with %6d  terms: RHS-LHS = %g\n',tst,mmax
end              % end of tst loop
% FILE d015.m ends.
```

Output:

```
 1. Test with   6495  terms: RHS-LHS = 2.47154
 2. Test with   9685  terms: RHS-LHS = 1.5223
 3. Test with   3870  terms: RHS-LHS = 1.72901
 4. Test with   5126  terms: RHS-LHS = 1.1328
 5. Test with   1210  terms: RHS-LHS = 1.24614
 6. Test with   4067  terms: RHS-LHS = 2.17711
 7. Test with   1346  terms: RHS-LHS = 1.50168
 8. Test with   9335  terms: RHS-LHS = 1.51314
 9. Test with  10562  terms: RHS-LHS = 2.11634
10. Test with   8829  terms: RHS-LHS = 1.0773
```

6. Consider a data set $(x_i, y_i), i = 1, ..., n$. We define $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

$$\begin{aligned} ss_{xx} &= \sum(x_i-\overline{x})^2 &&= \sum x^2 - n\overline{x}^2 \\ ss_{yy} &= \sum(y_i-\overline{y})^2 &&= \sum y^2 - n\overline{y}^2 \\ ss_{xy} &= \sum(x_i-\overline{x})(y_i-\overline{y}) &&= \sum xy - n\overline{x}\overline{y}. \end{aligned}$$

Write a MATLAB program that computes the correlation coefficient
the data set, defined as

$$r^2 = \frac{ss_{xy}^2}{ss_{xx}ss_{yy}}.$$

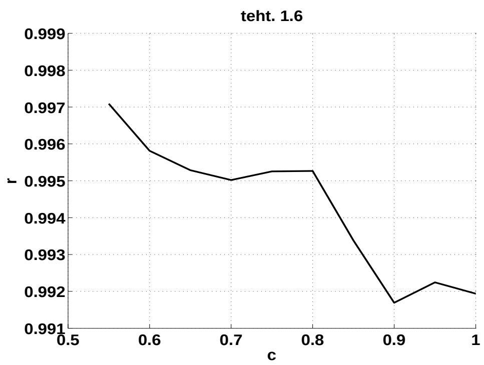
Create a syntetic data $x_i = i * 0.1, y_i = 0.7 * x_i + c * error_i$ where er
is uniformly distributed in $(-0.1, 0.1)$ with mean 0. One expects that
correlation coefficient decreases when c increases from 0.5 to 1. Check
with MATLAB.

Solution:

```
% FILE d016.m begins.
% MME08 harj 1 teht 6
close all
n =30;   %Fix n
x= (1:n)*0.1;
%myerror=(rand(1,n)-0.5)*0.3;
ssxx= sum((x-mean(x)).^2);
myval=[];
for tst=1: 10     % counter for test number
c=0.5+0.05*tst;
myerror=(rand(1,n)-0.5)*0.3;
y= 0.7*x+ c*myerror;
ssyy= sum((y-mean(y)).^2);
ssxy= sum((x-mean(x)).*(y-mean(y)));
r= ssxy/sqrt(ssxx*ssyy);
fprintf('%2d. test, c= %5.3f , r= %6.4f\n',tst, c, r)
     myval=[myval; [c r]];
end              % end of tst loop
figure
axes('FontSize',[18],'FontWeight','bold');
hold on
plot(myval(:,1), myval(:,2),'k-','LineWidth',2)
grid on
title('teht. 1.6','FontSize', 18,'FontWeight','bold');
xlabel('c')
ylabel('r')
% FILE d016.m ends.
```

Output:

```
 1. test, c= 0.550 , r= 0.9971
 2. test, c= 0.600 , r= 0.9958
 3. test, c= 0.650 , r= 0.9953
 4. test, c= 0.700 , r= 0.9950
 5. test, c= 0.750 , r= 0.9953
 6. test, c= 0.800 , r= 0.9953
 7. test, c= 0.850 , r= 0.9934
 8. test, c= 0.900 , r= 0.9917
 9. test, c= 0.950 , r= 0.9922
10. test, c= 1.000 , r= 0.9919
```

Solutions in Python

Problem 1.

```
# FILE d011.py begins
from Numeric import *
x,a=1,3.
print "    n     x(n)            Error\n"
for j in range(1,7):
    print "   %2d%16.10f   %12.4e" % (j-1,x,x-3**0.5)
    x=0.5*(x+a/x)
# FILE d011.py ends
```

Output:

```
    n     x(n)             Error

    0    1.0000000000    -7.3205e-01
    1    2.0000000000     2.6795e-01
    2    1.7500000000     1.7949e-02
    3    1.7321428571     9.2050e-05
    4    1.7320508100     2.4459e-09
    5    1.7320508076     0.0000e+00
```

Problem 2.

```
# FILE d012.py begins
from Numeric import *
s=0
```

```
print "%3s  %12s  %12s\n" % ('n', 'p(n)', 'Error')
for k in range(0,11):
    print "%3d  %12.8f  %12.4e" % (k,s,s-pi)
    s=s+(16.0**(-k))*(4.0/(8.0*k+1)-2.0/(8.0*k+4)-\
      1/(8.0*k+5)-1/(8.0*k+6))
# FILE d012.py ends
```

Output:

```
  n         p(n)         Error

  0    0.00000000   -3.1416e+00
  1    3.13333333   -8.2593e-03
  2    3.14142247   -1.7019e-04
  3    3.14158739   -5.2632e-06
  4    3.14159246   -1.9602e-07
  5    3.14159265   -8.1295e-09
  6    3.14159265   -3.6171e-10
  7    3.14159265   -1.6912e-11
  8    3.14159265   -8.2023e-13
  9    3.14159265   -4.0856e-14
 10    3.14159265   -1.7764e-15
```

Problem 3.

```
# FILE: d013.py begins
import os
from Numeric import *

def log2(x): return log(x) / log(2)

def h013f(x):
   n=floor(log2(x))
   y=pow(2.0,-n)*x*x+ pow(2.0,(n+1.0))
   return y

m=10.0
x= arange(pow(2.0,-m),1.0,0.01)
y=h013f(x)
fp=open('d013.dat','w')
for i in range(len(x)): fp.write("%f %f\n" % (x[i],y[i]))
fp.close()
```



```
fp=open('gnuplot.cmd','w')
fp.write("plot 'd013.dat' w l lw 2\npause -1")
fp.close()
os.system('gnuplot gnuplot.cmd')
# FILE: d013.py ends
```

Problem 4.

```
# FILE d014.py begins
from Numeric import *
from scipy import *
from mmeutil import *
from math import *
for j in [3,4,6]:
    x=0.*zeros(j+1)
    y=0.*zeros(j+1)
    for i in range(0,j):
        x[i]=cos(i*(2.*pi)/j)
        y[i]=sin(i*(2.*pi)/j)
    x[j]=1.
    y[j]=0.
    gplt.plot(x,y,'notitle w l')
    # School geometry: divide the figure into n
    # triangles, all with center angles 360/n
    schoolarea=j*0.5*sin(2*pi/j);
    formarea=0;
    for i in range(1,len(x)):
        formarea=formarea+0.5*(x[i-1]*y[i]-x[i]*y[i-1])
    print "%d -kulmio" % j
    print "Koulumatikka: %f  Kaava: %f\n" % (schoolarea,forma
 # FILE d014.py ends
```

Output:

```
3 -kulmio
Koulumatikka: 1.299038  Kaava: 1.299038

4 -kulmio
Koulumatikka: 2.000000  Kaava: 2.000000

6 -kulmio
```

```
Koulumatikka: 2.598076  Kaava: 2.598076
```

Problem 5.

```
# FILE d015.py begins.
# MME05 harj 1 teht 5
# Use random numbers such that the RHS series converge
# Choose mmax terms for each sum, mmax large
# Use terms a_k = rand *1/k, b_k = rand *1/k

from Numeric import *
import random

def rand(n):
   xx=1.0*arange(n)
   for i in range(n):
      xx[i]=random.uniform(0.1,1.0)
   return xx

for tst in range(1,11):     # counter for test number
   mmax=100+random.randint(0,1000)
   a= arange(1.0,mmax+1)
   a=(1.0/a)*rand(mmax)
   b= arange(1.0,mmax+1)
   b=(1.0/b)*rand(mmax)
   rhs1=sum(a*a)
   rhs2=sum(b*b)
   lhs=0           #The values of the left hand side
                    # will be accumulated in this variable
   for ii in range(mmax):
      for jj in range(mmax):
         lhs= a[ii]*b[jj]/(ii+jj +1)
   rhs=pi*sqrt(rhs1*rhs2) # The right hand side of the inequality
   print '%2d. Test with %6d  terms: RHS-LHS = %g' %(tst,mmax,rhs-lhs)
# FILE d015.m ends.
```

Output:

```
 1. Test with   1058  terms: RHS-LHS = 2.96545
 2. Test with    260  terms: RHS-LHS = 1.67632
 3. Test with    729  terms: RHS-LHS = 2.53008
```



```
 4. Test with    673  terms: RHS-LHS = 0.990683
 5. Test with    184  terms: RHS-LHS = 3.52681
 6. Test with    594  terms: RHS-LHS = 2.02634
 7. Test with    380  terms: RHS-LHS = 2.81594
 8. Test with    643  terms: RHS-LHS = 1.46423
 9. Test with    241  terms: RHS-LHS = 1.6612
10. Test with    356  terms: RHS-LHS = 2.23887
```

Problem 6.

```
# FILE d016.py begins.
# MME05 harj 1 teht 6
import os
import random
from Numeric import *

def rand(n):
   xx=1.0*arange(n)
   for i in range(n):
      xx[i]=random.uniform(0.1,1.0)
   return xx
def mean(x): return sum(x)/len(x)

n =30   # Fix n
x= arange(1.0,n+1.0)*0.1
ssxx= sum(pow(x-mean(x),2.0))
fname="d016.dat"
fp=open(fname,'w')
for tst in range(10):     # counter for test number
   c=0.5+0.05*tst
   y= 0.7*x+ c*(rand(n)-0.5)*0.2
# (rand(1,n)-0.5)*0.2 is a number in (-0.1, 0.1)
   ssyy= sum(pow(y-mean(y),2.0))
   ssxy= sum((x-mean(x))*(y-mean(y)))
   r= ssxy/(ssxx*ssyy)
   print '%2d. test, c= %5.3f , r= %6.4f' %(tst, c, r)
   fp.write('%f %f\n' % (c,r))
fp.close()

fp=open('gnuplot.cmd','w')
fp.write("plot 'd016.dat' w l lw 2\npause -1")
fp.close()
```

```
os.system('gnuplot gnuplot.cmd')
# FILE: d016.py ends
```

Output:

```
 0. test, c= 0.500 , r= 0.0633
 1. test, c= 0.550 , r= 0.0626
 2. test, c= 0.600 , r= 0.0625
 3. test, c= 0.650 , r= 0.0639
 4. test, c= 0.700 , r= 0.0638
 5. test, c= 0.750 , r= 0.0650
 6. test, c= 0.800 , r= 0.0630
 7. test, c= 0.850 , r= 0.0630
 8. test, c= 0.900 , r= 0.0642
 9. test, c= 0.950 , r= 0.0631
```