City University of Hong Kong

AP Courses Review Notes

AP3114

Computational Methods for Physicists and Materials Engineers

Version 1.0

February 28, 2018

All LaTeX (`.tex`) files of this document can be accessed from https://github.com/zzw42/review-notes-cityu.

Contents

Preface **v**

1 Typical Problems in Physics and Materials Engineering & Introduction to Scilab **1**
1.1 Homework 1

2 Matrix **4**
2.1 Lecture 4
2.2 Homework 4

3 Basic Programming **7**
3.1 Lecture 7
3.2 Homework 8

4 Program II & Plotting I **12**
4.1 Lecture 12
4.2 Homework 13

5 Programming III & Plotting II **16**
5.1 Lecture 16
5.2 Homework 18

6 Random Number **21**
6.1 Lecture 21
6.2 Homework 22

7 Statistical Description and Analysis of Data **26**
7.1 Lecture 26
7.2 Homework 26

8 Solving Non-linear Equations **28**
8.1 Lecture 28
8.2 Homework 31

10 Solving Differential Equations I **36**
10.1 Lecture 36
10.2 Homework 37

11 Solving Differential Equations II **43**
11.1 Lecture 43

12 Fourier Series **45**
12.1 Lecture 45
12.2 Homework 45

A Mid-Term Exam **47**

B Final Exam **51**

Preface

The first few exercises are from *Introduction to Scilab* by Michaël Baudin. The homepage of this document is http://forge.scilab.org/index.php/p/docintrotoscilab/, where the source can also be found.

The note on cubic equation by Urs Oswald is also helpful. http://www.ursoswald.ch/download/CUBIC.pdf.

These `.sci` files have been tested under Scilab 5.52.

Lecture 1

Typical Problems in Physics and Materials Engineering & Introduction to Scilab

1.1 Homework

```
//PART 1

//Precedence
2*3+4
2+3*4
2/3+4
2+3/4
//Parentheses
2*(3+4)
(2+3)*4
(2+3)/4
3/(2+4)
//Exponents
1.23456789d10
1.23456789e10
1.23456789e-5

//PART 2
//Functions
sqrt (4)
sqrt(9)
sqrt(-1)
sqrt(-2)
sqrt(1)
log(exp(2))
exp(log(2))
10^2
log10(10^2)
10^log10(2)
```

```
sign(2)
sign(-2)
sign(0)

//Trigonometry
cos(0)
sin(0)
cos(%pi)
sin(%pi)
cos(%pi/4)-sin(%pi/4)

a=rand(2,3);
typeof(a)
a=[a,zeros(2,1)]
a='scilab';typeof(a)
exists('a')
clear('a');exists('a')
a= "Scilab"
b=rand(2,2)
b= b>=0.5
L=list(a,b)
A.x=32;A.y=%t
a=spec(rand(3,3))

//PART 4
S="a string with a quote character<<"'>>"
S='a long string 0...
using continuation'
S=['A','string';'2x2','matrix']
length(S)
~(1>=2)
%t & %t

//PART 5
x=-10:0.1:10;
y=((x>=0).*exp(-x))+((x<0).*exp(x));
y=bool2s([%t,%f])
%e
%pi
2+3*%pi
disp("Bob won")
d=500;
disp("Bob won "+string(d)+" dollars")
disp("It''s fair")

//PART 6
a=sqrt(3)
b=1
c=2
a^2+b^2==c^2
abs(a^2+b^2-c^2)<%eps
abs(a^2+b^2-c^2)/c^2<%eps

// PART 7
// a (might be wrong) explainition
// when boolean operators (&, |, ~) deal with constant
// 0 represents %f, all others represent %t
```

```
1 & -1
13 & ~(-6)
0 < -2|0
~[1 0 2] * 3
5 > 4 > 3
2 > 3 & 1

//PART 8
a=[1 0 2]
b=[0 2 2]
a ~= b
a < b
a<b<a
a<b<b
a | (~a)
b& (~b)
a(~(~b))
a=b==a
a=b==a
```

./personal_answer_Homework/lecture1_exe.sce

Lecture 2

Matrix

2.1 Lecture

```
A =[1,2,3;4,5,6]
disp("Matrix A is ")
disp(A)
A =[1,2,3
4,5,6]
clear;
//A = ones(100,100)
A(3,1) = 7
A(:,3)=[]
```

./lecture/lecture_2_matrix.sce

2.2 Homework

```
//PART 1
//Plus one
x = 1:4;
y = x + 1
//Vectorized multiplication
x = 1: 4;
y = 5 : 8;
z = x .* y
//Vectorized invert
x = 1 : 4;
y = 1 ./ x



//PART 2
//Vectorized division
x = 12*(6:9);
```

```
y = 1:4;
z = x ./ y
//Vectorized squaring
x = 1 : 4;
y = x.^2
//Vectorized sinus
x = linspace (0,%pi ,10);
y = sin(x)
//Vectorized function
r = 2.220D -16;
x = linspace ( -16 ,0 ,10);
y = log10 (r ./10.^ x + 10.^ x);




//PART 3
A=[1,2,3+5]
A=[1,2,3*5]
A=[A,0;1,2,3,4]
A=[eye(2,1),3*ones(2,3);linspace(3,9,4);zeros(1,4)]
d=diag(A)'
B=diag(d)
C=matrix(d,2,2)




//PART 4
A=rand(2,2);
B=exp(A)
B=expm(A)
clear A;
A(2,4)=1
A([1,2],[1,2])=int(5*rand(2,2))
A([1,2],[1,3])=[]
A(:,1)=8
A(:,$)=[]
A(:,$+1)=[4;5]
A=int(10*rand(3,7));
B=A([1,3],$-1:$)




//PART 5
A=(1:3)'*ones(1,3)
A.*A'
t=(1:3)';m=size(t,'r');n=3;
A=(t*ones(1,n+1)).^(ones(m,1)*[0:n])
A=eye(2,2).*.[1,2;3,4]
A=[1,2;3,4];b=[5;6];
x=A\b;norm(A*x-b)
A1=[A,zeros(A)];x=A1\b
```

```
A1=[A;A];x=A1\[b;7;8]
77
79
81
//PART 6
83 A=rand(2,8,'n');
A=sign(A);
85 A=string(A)
A=strsubst(A,'1','+');
87 A=strsubst(A,'-+','-')
89 name='x';n=3;val=[45,67,34];
str=name+string(1:n)+'=val('+string(1:n)+');'
91 execstr(str);
[x1,x2,x3]
93
95
97 //PART 7
A= int(10*rand(1,7))
99 A(A>=3) = 0
I=find(A == 0)
101 A=sprand(100,100,0.1);
whos('-type','sparse')
103 B=full(A);
whos('-name','B');
105 timer();inv(B);timer()
timer();inv(A);timer()
107
109
111
//PART 8
113 x.color = 4;
x.value = rand(1,3);
115 x.name = 'foo';
x
117 r = 1/%s
a=[1,r;1,1]
119 b=inv(a)
b.num
121 b.den
sys=ssrand(1,1,2)
123 sys.A
L=list()
125 L(2)=testmatrix('mag1',3)
L(0)=34
127 L($+1)='Y'
[a,b]=L([1,3])
129 L(2)=null();
```

./personal_answer_Homework/lecture2_exe.sce

Lecture 3

Basic Programming

3.1 Lecture

```
// codes in lecture 3

//#1: if
i = 1;

if (i == 1 ) then
    disp("Hello!")
elseif( i == 2 ) then
    disp("Goodbye!")
elseif( i == 3) then
    disp("Thao !")
else
    disp("what")
end


//#2: case
i = 2
select i
case 1
    disp("one")
case 2
    disp("two")
case 3
    disp("three")
else
    disp("ha")
end


//#3: 'for' loop
for i = 1:5
    disp(i)
end
```

```
for i = 0.5:2;5
    disp(i)
end


//#4: 'while' loop
s = 0
i = 0
while(i <= 10)
    s = s + i
    i = i +1
end
disp(s)


// #5: using 'continue' in a loop
s = 0
i = 0

while (i < 10 )
    i =i + 1
    if( modulo (i, 2)== 0 ) then
        continue
// continue is the command in `for` or `while` "loop"
// to skip the lines between continue and end of the *loop*
    end
    s = s + i
end


// #6: example: caculating balance
balance = 1000
year = 2017
interest = 0.2
while( balance <= 2000 )
    balance = balance * (1+interest)
    year = year + 1
end
disp ('The year is ' + string(year))
disp ('The balance is ' + string(balance))


// #7: using 'pause' to debug
function y=mysum(istart,iend)
    pause
    y = sum( istart: iend)
endfunction
```

./lecture/lecture_3_basic_programming.sce

3.2 Homework

```
//PART 1
v = [3;-2;5]
v = [3, -2, 5]
```

```
m = [1 2 3; 4 5 6; 7  8 9]
m = [1 2 3; 4 5 6]
m(2,3)
m(2,:)
m(:,3)
m'


//PART 2
A = [1, 2, 3; 4, 5, 6]
B = [1;1;2]
A * B
A * A
A .* A
2*(A+2)
A/A
A./A
C=1:4
C*C
C.*c
1/C
(1)./C


// PART 3
A = [1 2 3; 4 5 6]
B = [1; 1]
X =A\B
v = [2,6,9,6,-4,0,2]
gsort(v,"g","i")
gsort(v)
unique(v)
U=[1:10]
length(U)
sum(U)
prod(U)
m=[1 2 3; 4 5 6];
size(U)


//PART 4
w=[1,5,3,8,14,7,3,2,12,6];find(w<5)
w=[1,5,3,8,14,7,3,2,12,6];find(w==3)


//PART 5
true = %t
if true then
     disp("hello"),
end

A=log(rand(3,3));
if imag(A)==0 then
    disp ('A is a real number');
end

if  imag(A)==0 then
```

```
    disp ('A is a real number');
else
    disp ('A is complex');
end


//PART 6
n=89;
isprime=%t
for i=2:(n-1)
    if pmodulo (n,i)==0
        then isprime=%f;
        break;
    end
end
isprime


n=16778;
timer();
res=[];
for i = 2:(n-1)
    if pmodulo (n,i) == 0 then
         res = [res,1];
    end
end
t1 = timer();

res
v = 2:(n-1);
timer();
I = find(pmodulo (n,v)==0);
res = v(I)
t2 = timer();

[t1,t2]


//PART 7
x=1;
while exp(x)<>%inf;
    x = x+1;
end
[exp(x-1),exp(x)]==%inf

x=[1:3];
while exp(x)<>%inf;
    x = x+1;
end
exp(x)==%inf

x=1;
while %t
    if exp(x)==%inf then
        break;
    end
    x=x+1;
```

```
118 end
    [exp(x-1),exp(x)]==%inf
120

122 //PART 8
    function y=foo(x,g);
124     y=g(x);
    endfunction
126 typeof(foo)
    foo(%pi,sin)
128 foo(%pi,sinh)==sinh(%pi)
    v=rand(1,10);
130 foo(3,v)

132 function B=f(A)
        B=string(sign(A));
134     B=strsubst(strsubst(B,'1','+'),'-+','-')
    endfunction
136 f(rand(2,5,'n'))
```

./personal_answer_Homework/lecture3_exe.sce

Lecture 4

Program II & Plotting I

4.1 Lecture

```
// #1: defining function
function y=myfunction(x)
    y=2*x
endfunction

// #2: function(3)
function y=myfunction(x)
    z=2*x
endfunction

// #3: function(4) multiple output
function [y1,y2]=simplef(x1,x2)
    y1 = 2 * x1
    y2 = 3 * x2
endfunction

// #4: call anonther function
function y=fmain(x)
    y=2*flevel1(x)
endfunction

function y=flevel1(x)
    y=2*flevel2(x)
endfunction

function y=flevel2(x)
    y=2*x
    whereami
endfunction

// example: detering tax
function y=revenue(x)
    if  x<=10000 then
        y=0.1*x
    elseif x<=20000 then
```

```
36          y=1000+0.2*(x-20000)
    elseif x<=40000 then
38          y=3000+0.3*(x-40000)
    else
40           y=9000+0.5*(x-40000)
    end
42 endfunction

44 // HW3 Exercise 6
n=89;
46 isprime=%t
for i=2:(n-1)
48      if pmodulo (n,i)==0 then
            isprime=%f;
50          break;
        end
52 end
isprime
```

./lecture/lecture_4_programmingii_plottingi.sce

4.2 Homework

```
1  //PART 1
   function y=fact(x)
3      if x<=1 then
           y=x;
5      else
           y=x*fact(x-1);
7      end
   endfunction
9  fact(4)

11 function y=f(x)
       y=2*x
13 endfunction

15 x=90;
   f()
17
   f(5,7)
19 [a,b]=f(5)

21 //PART 2
   function y=f(x);
23     z=x;
   endfunction
25 y=89;
   z=67;
27 w=f(x)

29 function y=f();
       y=x;
31 endfunction
   x=5;
```

```
y=f()

function y=f();
    x=2*x;
    y=x;
endfunction
y=f()


x=[56,67];
function y=f();
    x(1)=5;
    y=x
endfunction
y=f()

//PART 3
function d=dollars(e,t);
    d=e*t;
endfunction
dollars(200,1.4)

function y=f(x);
    y=36/(8+exp(-x));
endfunction

function y=g(x);
    y=4*x/9+4;
endfunction
f(10)
g(12.5)

//PART 4
u(1)=4;
for i=1:19
    u(i+1)=u(i)+2*i+3;
end
disp(u)

year=2005;
height=120;
while(height <700);
    height=height+30;
    year=year+1;
end
disp(year)

//PART 5
y=[length(find(w<0)) length(find(w==0)) length(find(w>0))];
w=[-4 0 5 -3 0 3 7 -1 6]
disp(y)

//PART 6
function y=cost(x)
    if  x <= 500 then
        y = 0.02 * x
    elseif x<=1000 then
```

```
        y=10+0.05*(x-500)
    else
        y=35+0.1*(x-1000)
    end
endfunction
y = [cost(200) cost(500) cost(700) cost(1000) cost(1500)]
//y = cost([200 500 700 1000 1500]) //not working

//PART 7

function y=win(Y)
if Y==[6 6 6] then
    y=20
    elseif length(unique(Y))==1 then
    y=10
    elseif length(unique(Y))==2 then
    y=5
    else
    y=0
end
endfunction

s=0;
for i=1:1000
    Y = grand(1, 3, "uin", 1, 6);
    w=win(Y);
    s=s+w;
end
disp(s/1000)
```

./personal_answer_Homework/lecture4_exe.sce

Lecture 5

Programming III & Plotting II

5.1 Lecture

```
// plotting
for n=1:50
    u(n)=(-0.8)^n;
end

clf; plot(u,"*r")


// function 'f'
function y=f(x)
    y=(x^2+2*x)*exp(-x)
endfunction

x=linspace(-2,5,50);
plot(x,f)

function y=f(x)
    y=(x^2+2*x)*exp(-x)
endfunction
function y=g(x)
    y=sin(x/2)
endfunction

x=linspace(-2,5,50);
clf
plot(x,f,"r",x,g,"g")


function f=myquadratic(x)
    f=x.^2
endfunction
xdata= linspace(1,10,50);
```

```
ydata= myquadratic(xdata);
plot(xdata, ydata)
xtitle("Diagram","Year","Income")


function f=myquadratic(x)
    f=x.^2
endfunction

function f=myquadratic2(x)
    f=2*x.^2
endfunction
xdata= linspace(1,10,50);
ydata= myquadratic(xdata);
plot(xdata, ydata,"+-")
ydata2= myquadratic2(xdata);
plot(xdata, ydata2,"o-")
xtitle("Diagram","Year","Income")
legend("x^2","2x^2")
xs2png(0,"h.png")

// clear all data
xdel(winsid());
clear;
clc;


//Figure #1: Basic plot with LaTeX annotations
//-----------------
//Data
x = linspace(-5,5,51);
y = 1 ./(1+x.^2);

//Plot
scf(1);
clf(1);
plot(x,y ,'o-b');
xlabel("$-5 \le x \le 5 $","fontsize",4,"color","red");
ylabel("$y(x)=\frac{1}{1+x^2}$", "fontsize",4,"color","red");
title("Range function (#Points= "+ string(length(x)) + ").","color"
    ,"red","fontsize",4);
legend("Function evaluation")


//Figure #7: Subplot with real and imaginary part
//---------------------------
//Data
t=linspace(0,1,101);
y1=exp(%i*t);
y2=exp(%i*t.^2);
//Plot
scf(7);
clf(7);
subplot(2,1,1); //2*1 matrix with no. 1
plot(t,real(y1),'r');
plot(t,real(y2),'b');
xtitle("Real part");
```

```
subplot(2,1,2); //2*1 matrix with no. 2
plot(t,imag(y1),'c');
plot(t,imag(y2),'pm');
xtitle("Imaginary part");


//Figure #11 : Surface with a color map
x = -1:0.1:1;
y = -1:0.1:1;
[X,Y]=meshgrid(x,y);
Z=X .^3 +Y.^2;
//Plot
scf(14);
clf(14);
xset("colormap",jetcolormap(64));
surf(X,Y,Z);
xlabel('X');ylabel('Y');zlabel('Z');
```

./lecture/lecture_5_programmingiii_plottingii.sce

5.2 Homework

```
//Exercise 1
X=[1,3,3,7,7,9,10];
Y=[8,7,5,5,4,2,2];
scf(1);
clf(1);
figure(1);
plot(X,Y,"*r")
xs2png(1,"5_1.png")
```

./personal_answer_Homework/lecture5_exe1.sci

```
//Excercise 2
x=linspace(-4* %pi,4 *%pi,1000);
y=3* sin(x)./x +cos(x);
scf(2);
clf(2);
figure(2);
plot(x,y)
xs2png(2,"5_2.png")
```

./personal_answer_Homework/lecture5_exe2.sci

```
function y=free(t)
    y=H-(1/2) .* g .* t.^2
endfunction

function y=resistance(t)
    y=H-(g/k) .*t + (g/k^2) .* (1-%e .^(-k*t))
endfunction

H=495;
```

```
g=9.8;
k=0.1;
t=linspace(0,10,100);

scf(3);
clf(3);
figure(3);
plot(t,free(t),"r",t,resistance(t),"b");
xlabel("Time(s)");
ylabel("Distance(m)");
legend(["Free fall ","Resistance"])
xs2png(3,"5_3.png")

t = fsolve(10,free);
disp(t);
```

./personal_answer_Homework/lecture5_exe3.sci

```
function y=f(x)
    y=(x.^2 + 2.*x).*( %e.^(-x) )
endfunction

function y=g(x)
    y=sin(x./2)
endfunction

function y=h(x)
    y=f(x)-g(x)
endfunction

x=linspace(-2,5);

scf(4);
clf(4);
figure(4);
plot(x,f(x),"r",x,g(x),"b");
legend(["$f(x)$","$g(x)$"])
xs2png(4,"5_4.png");


a(1) = fsolve(-2,h);
a(2) = fsolve(-1,h);
a(3) = fsolve(2,h);
disp(a);
```

./personal_answer_Homework/lecture5_exe4.sci

```
theta=0: 0.01: 5*%pi;
r1= theta;
r2=2 * theta;

scf(5);
clf(5);
polarplot(theta,r1,[2,2]);
polarplot(theta,r2,[5,2]);
xs2png(5,"5_5.png")
```

./personal_answer_Homework/lecture5_exe5.sci

```
function y=f(x)
    y=(1/8)* x.^2-8
endfunction


scf(6);
clf(6);
figure(6);
x=linspace(-8,8);
y=linspace(0,2* %pi);
plot(x,f(x),"r")
plot(10*cos(y),10*sin(y))
plot(-4,4,'pr','MarkerSize',12);
plot(4,4,'pr','MarkerSize',12);
legend("mouth","head","eyes")
title("MatLab Art")
xs2png(6,"5_6.png");
```

./personal_answer_Homework/lecture5_exe6.sci

Lecture 6

Random Number

6.1 Lecture

```
t=linspace(0,4*%pi,100);
param3d(cos(t),sin(t),t)

//t=linspace(-20*%pi,20*%pi,2000);
//param3d1(sin(t),t.*cos(t)/max(t),t/100)


//x=linspace(-%pi,%pi,40);
//y=linspace(-%pi, %pi, 40);
//plot3d(x,y,sinh(x')*cos(y));


// #1: leap year
if modulo(year,4)<>0 then
    disp("Not leap year!")
elseif modulo(year,100)<>0 then
    disp("Leap Year!")
elseif modulo(year,400)<>0 then
    disp("Not Leap Year!")
else
    disp("Leap Year!")
end

// #2: Inflation
clear;
number=200;
inflation=5.6;
year=4;
price=1;

capital = number * price;
cost=0;

i=0;
while i<year
```

```
    price = price * (1+inflation/100);
    i = i+1
    cost = cost + number * price;
end
disp(cost)

new_number = int(capital / price);
disp(new_number)

// #3: Adding-up numbers
clear;
n=12349;
remainder=0;
i=1;
while %t
    if modulo(n,i)==n then
        break;
    end
    i = i*10;
    remainder = remainder + modulo(n,i)/i*10;
    n = n - modulo(n,i);
end
disp(remainder)

// #4: Vending machine
br = 100;
coupon = 100;

while coupon > 7
    br = br + int (coupon ./ 7)
    coupon = modulo(coupon, 7) + int (coupon ./ 7)
end
disp(br,coupon)


// Returns a 400 by 800 matrix of random doubles
R = grand(400,800,"def")
scf(1);
clf(1);
histplot(10,R); //10 classes(blocks)
xtitle("Uniform distribution", "X", "Frequency")

// Normal distribution
R = grand(400,800,"nor", 0, 1 )
scf(2);
clf(2);
histplot(20,R); //10 classes(blocks)
xtitle("Normal distribution", "X", "Frequency")
```

./lecture/lecture_6_random–number.sce

6.2 Homework

```
x = -2.1:0.1:2.1;
y = -6:0.1:6;
```

```
[X Y]=meshgrid(x,y);
Z = 80 * (Y.^2) .*(exp(-X.^2-0.3 * Y.^2));
//Plot
scf(1);
clf(1);
figure(1);
xset("colormap",jetcolormap(64));
surf(X,Y,Z);
xs2png(1,"6_1.png")
```

./personal_answer_Homework/lecture6_exe1.sci

```
x = linspace(-%pi,%pi,50);
y = linspace(-%pi,%pi,50);
figure(1);
plot3d(x,y,sin(x')*cosh(y));
xs2png(1,"6_2.png")
```

./personal_answer_Homework/lecture6_exe1b.sci

```
//n=11;
//P=zeros(n);
//
//for i= 2:11;
//    P(i-1,i)=0.6;
//    P(i,i)=0.4;
//end
//
//P(1,1)=1;
//x = [0 0 0 0 0 0 0 0 0 0 1]';
//
//for t=1:600
//    x = P * x;
//    disp ([t x'])
//end


function y=go()
    Y = grand(1, 1, "def");
    if Y > 0.4 then
        y = s + 1;
    else
        y = s - 1;
    end
endfunction

s = 0;
for i=1:600
    if s < 10 then
      s = go();
    else
      break;
    end
end

disp(s)
```

./personal_answer_Homework/lecture6_exe2.sci

```
function Y=find(N)
    Y = grand (1, 1, "bin", N, 0.8)
endfunction

y(1) = find(4);
y(2) = find(2*y(1));
disp(y)
```

./personal_answer_Homework/lecture6_exe3.sci

```
A=1:99;A=A'
for i=1:10000
    x = grand(1, 1, "uin", 1, 99) ;
    y = grand(1, 1, "uin", 1, 99) ;
    A([x y],:)=A([y x],:)
end
disp(A)
```

./personal_answer_Homework/lecture6_exe4.sci

```
//the two-engined DFII
function y=testA()
        X = grand(1, 1, "def");
        Y = grand(1, 1, "def");
    if Y > 0.5 & X > 0.5 then
        y = 1;
    else
        y = 0;
    end
endfunction


A=0;
for i=1:1000
    A=A+testA();
end
disp(A/1000)

//four-engined DFIV
function y=testB()
        X = grand(1, 1, "def");
        Y = grand(1, 1, "def");
        A = grand(1, 1, "def");
        B = grand(1, 1, "def");
    if X > 0.5 & Y > 0.5 & A > 0.5 then
        y = 1;
    elseif X > 0.5 & Y > 0.5 & B > 0.5 then
        y = 1;
    elseif X > 0.5 & A > 0.5 & B > 0.5 then
        y = 1;
    elseif Y > 0.5 & A > 0.5 & B > 0.5 then
        y = 1;
```

```
33      else
            y= 0;
35      end
    endfunction
37
    B=0;
39  for i=1:1000
        B=B+testB();
41  end
    disp(B/1000)
```

./personal_answer_Homework/lecture6_exe5.sci

Lecture 7

Statistical Description and Analysis of Data

7.1 Lecture

```
//Summary of sample statistics
M = int(100*rand(20,4))
mean(M,'r')
mean(M,'c')
//median(M,'r')
//stdev(M,'r')
//mean(M)
//median(M)
//stdev(M)

// covariance and correlation
x = 5 * rand(1,10);
y = 5 * rand(1,10);
sx = stdev(x);
sy = stdev(y)'
[sxy meanxy] = corr (x,y,1) //covariance, mean of x and y
rxy = sxy/ (sx * sy) //correlation
```

./lecture/lecture_7_statistics.sce

7.2 Homework

```
//Lecture6 exercise 1
M = int(100*rand(10,6))
a=mean(M)
b=median(M)
c=stdev(M)
g=mean(M,'c')
h=median(M,'c')
```

```
i=stdev(M,'c')
d=mean(M,'r')
e=median(M,'r')
f=stdev(M,'r')
disp(a);
disp(b);
disp(c);
disp(d);
disp(e);
disp(f);
disp(g);
disp(h);
disp(i);
```

./personal_answer_Homework/lecture7_exe1.sci

```
//Lecture6 excercise 2
A = [2 51  39 24 50 42 8 62 34 70 52 28 65 8]
B = [96 50 52 55 56 46 9 98 81 42 24 92 10 46]
Sab=corr(A,B,1)
Sa=stdev(A)
Sb=stdev(B)
r=Sab/(Sa*Sb)
disp("Correlation coefficient is "+string(r))
```

./personal_answer_Homework/lecture7_exe2.sci

Lecture 8

Solving Non-linear Equations

8.1 Lecture

```
// #1: Roots of quadrotic equation
a=1;
b=1;
c=2;
D = b^2-4*a*c;

if D > 0 then
    x1 = (-b + D^(1/2)) / (2*a);
    x2 = (-b - D^(1/2)) / (2*a);
    disp(x1,x2);
elseif D==0 then
    x = -b / (2*a);
    disp(x);
else
    xr = -b / (2*a);
    xi = (-D)^(1/2) / (2*a);
    x1 = xr + xi *%i;
    x2 = xr - xi *%i;
    disp(x1,x2);
end

// #2: Roots of cubic equation
// from official solutions
function [root1, root2, root3] = cubicroots(d, a, b, c)
// First calculate p, q and D as distinguishers
a=a/d;
b=b/d;
c=c/d;
p = (3*b-a^2)/3;
q = c + 2*(a^3)/27 - a*b/3;
D = (q/2)^2 +(p/3)^3;

```

```
if D>0 then
// One real and two complex conjugate solutions
// Here we have to treat Scilab to take cubic root of a real number
     so it will not return the complex form
    u=sign(-(q/2)+sqrt(D))*abs(-(q/2)+sqrt(D))^(1/3);
    v=sign(-(q/2)-sqrt(D))*abs(-(q/2)-sqrt(D))^(1/3);
    root1=-(a/3)+u+v;
    realroot23=-(a/3)-((u+v)/2);
    imagroot23=3^(1/2)*(u-v)/2;
    root2=realroot23+(imagroot23*%i)
    root3=realroot23-(imagroot23*%i)
    disp("First root is "+string(root1));
    disp("Second root is "+string(root2));
    disp("Third root is "+string(root3));

elseif D==0
//Three real solutions with one double root
    u=-(q/2)+(sqrt(D))^(1/3);
    v=-(q/2)-(sqrt(D))^(1/3);
    root1=-(a/3)+u+v;
    root2=-(a/3)-((u+v)/2);
    disp("The first root is "+string(root1));
    disp("The other 2 solutions are a double root "+string(root2));

else
// Three distinct real solutions
// Using trigonometric form to calculate the roots
    theta=acos((-q)/(2*((abs(p)/3)^(3/2))));
    theta1=theta/3;
    theta2=(theta- 2 * %pi)/3;
    theta3=(theta+ 2 * %pi)/3;
    root1= -(a/3) + 2*((abs(p)/3)^(1/2))*cos(theta1);
    root2= -(a/3) + 2*((abs(p)/3)^(1/2))*cos(theta2);
    root3= -(a/3) + 2*((abs(p)/3)^(1/2))*cos(theta3);
    disp("First root is "+string(root1));
    disp("Second root is "+string(root2));
    disp("Third root is "+string(root3));
end
endfunction

//mprintf("We found roots for x^3+1=0: \n")
//cubicroots(1,0,0,1);
//mprintf("\n")
//mprintf("We found roots for 3x^3+5x^2+2x+6=0: \n")
//cubicroots(3,5,2,6);
//mprintf("\n")
//mprintf("We found roots for x^3+x^2+x+1=0: \n")
//cubicroots(1,1,1,1);
//mprintf("\n")
//mprintf("We found roots for 9x^3+3x^2+4x+6=0: \n")
//cubicroots(9,3,4,6);
//mprintf("\n")


// #3: Caculating non-linear eqations
function [x]=half(a,b,f)
//interval halving routine
```

```
N = 100; eps = 1.e-5; // define max. no. iterations and error
if (f(a)*f(b) > 0) then
    error('no root possible f(a)*f(b) > 0')
    abort;
end;

if(abs(f(a)) < eps) then
    error('solution at a')
    abort;
end;

if(abs(f(b)) < eps) then
    error('solution at b')
    abort;
end;

while (N > 0)
    c = (a+b)/2;
    if (abs(f(c)) < eps) then
        x = c;
        x
        return;
    end;

    if (f(a)*f(c) < 0) then
        b = c;

    else
        a = c;
    end;
    N = N - 1;
end;
error('No convergence')
abort;
//end function
endfunction

function [x]=newton(x0,f,fp)
//newton-raphson algorithm
N = 100; eps = 1.e-5; // define max. no. iterations and error
maxval = 10000.0; // define value for divergence
xx = x0;
while (N>0)
    xn = xx - f(xx)/fp(xx);
    if(abs(f(xn))<eps) then
        x = xn
        disp(100-N);
        return(x);
    end;
    if (abs(f(xx))>maxval) then
        disp(100-N);
        error('Solution diverges');
        abort;
    end;
    N = N - 1;
    xx = xn;
end;
```

```
error('No convergence');
abort;
//end function
endfunction

deff('[y]=hi(x)','y=x')
half(-1,0.5,hi)

// #4: roots of a complex number
z = -16;
r = abs(z);
theta = -%pi;
roots_of_z = [];
for k = 0:3
    roots_of_z = [roots_of_z r^(1/4)*exp(%i*((theta+2*k*%pi)/3))];
end;
roots_of_z
```

./lecture/lecture_8_9_solving_nonlinear_equations.sce

8.2 Homework

Exercise 1 Calculate all the cubic roots of below numbers in complex field:

1. 1
2. i
3. $5+3i$
4. $2e^{i\pi/2}$
5. $-2-3i$

```
disp('Exercise 1')
function y=complexroots(r, theta)
y = [];
for k = 0:2
y = [y r^(1/3)*exp(%i*((theta+2*k*%pi)/3))];
end;
endfunction
z1=1;
z2=%i;
z3=5 + 3*%i;
z4=2 * %e^((%i*%pi)/2)
z5=-2 - 3*%i;

r1=abs(z1);theta1=0;
r2=abs(z2);theta2=%pi/2;
r3=abs(z3);theta3=atan(imag(z3)/real(z3));
r4=abs(z4);theta4=%pi/2;
r5=abs(z5);theta5=atan(-3/-2);

disp(complexroots(r1,theta1))
```

```
disp(complexroots(r2,theta2))
disp(complexroots(r3,theta3))
disp(complexroots(r4,theta4))
disp(complexroots(r5,theta5))
```

./personal_answer_Homework/lecture8_exe1.sci

Exercise 2 Write a Scilab program `quadroots` to compute and print the roots of a quadratic equation $ax^2 + bx + c = 0$ according to the previous pseudo-code. It should run with a command like

`quadroots(1,3,2)`

for the case of $x^2 + 3x + 2 = 0$.

Test `quadroots` on the following examples, checking the results in each case.

1. $x^2 + 1 = 0$
2. $0x^2 + 2x + 1 = 0$
3. $x^2 + 3x + 2 = 0$
4. $4x^2 + 24x + 36 = 0$

```
disp('Exercise 2')
function y=quadroots(a,b,c)
D = b^2-4*a*c;
if a == 0 then
    x = -c/b;
    y = x
elseif D > 0 then
    x1 = (-b+D^(1/2))/(2*a);
    x2 = (-b-D^(1/2))/(2*a);
    y=[x1,x2];
elseif D==0 then
    x = -b/(2*a);
    y=x;
else
    xr = -b/(2*a);
    xi = (-D)^(1/2)/(2*a);
    x1 = xr + xi *%i;
    x2 = xr - xi *%i;
    y=[x1,x2];
end
endfunction

disp(quadroots(1,0,1))
disp(quadroots(0,2,1))
disp(quadroots(1,3,2))
disp(quadroots(4,24,36))
```

./personal_answer_Homework/lecture8_exe2.sci

Exercise 3 Write a Scilab program `cubicroots` to compute and print the roots of a quadratic equation $ax^3 + bx^2 + cx + d = 0$ according to the previous pseudo-code. It should run with a command like

`cubicroots(4,5,1,2)`

for the case of $4x^3 + 5x^2 + 1x + 2 = 0$.

Test `cubicroots` on the following examples, checking the results in each case.

```
disp('Exercise 3')
function y=cubicroots(d,e,f,g)
    a = e/d;
    b = f/d;
    c = g/d;
p = (3*b-a^2)/3;
q = c + (2*(a^3)/27)-(a*b)/3;
D = (q/2)^2 + (p/3)^3;

    u = sign(-q/2 + D^(1/2)) * (abs(-q/2 + D^(1/2)))^(1/3);
    // caution: when caculate
    // the cubic root of a negative number, it might
    // give the coomplex root
    v = sign(-q/2 - D^(1/2)) * (abs(-q/2 - D^(1/2)))^(1/3);
    xr = -(a/3) - (u+v)/2;
    xi = ((3)^(1/2)) * (u-v)/2;
if D > 0 then
    x1 = -(a/3) + u + v;
    x2 =  xr + %i* xi;
    x3 =  xr - %i* xi;
    y = [x1,x2,x3];

elseif D == 0 then
    x1 = -(a/3) + u + v ;
    x2 = -(a/3) -(u+v)/2;
    x3 = -(a/3) -(u+v)/2;
    y = [x1,x2,x3]

else
    cosphi = - q/(2 * (abs(p)/3)^(3/2))
    phi = acos(cosphi);
    phi1 = phi /3;
    phi2 = (phi-%pi)/3;
    phi3 = (phi+%pi)/3;
    x1= -a/3 + 2* (abs(p)/3)^(1/2)*cos(phi1);
    x2= -a/3 + 2* (abs(p)/3)^(1/2)*cos(phi2);
    x3= -a/3 + 2* (abs(p)/3)^(1/2)*cos(phi3);
    y = [x1, x2, x3]
end
endfunction


disp(cubicroots(1,0,0,1));
disp(cubicroots(3,5,2,6));
disp(cubicroots(1,1,1,1));
disp(cubicroots(9,3,4,6));
```

./personal_answer_Homework/lecture8_exe3.sci

Exercise 4 Find the roots of below polynomial using the function `roots`:

```
disp('Exercise 4')
p1 = poly([-10 1 0 5 0 3],'x','coeff')
p2 = poly([1 0 -1 0 0 1 0 -6],'x','coeff')
p3 = poly([3 1 -1 0 1],'x','coeff')
disp(roots(p1))
disp(roots(p2))
disp(roots(p3))
```

./personal_answer_Homework/lecture8_exe4.sci

Exercise 5

```
disp('Exercise 5')

function [x]=half(a,b,f)
//interval halving routine
N = 100; eps = 1.e-4; // define max. no. iterations and error
if (f(a)*f(b) > 0) then
    error('no root possible f(a)*f(b) > 0')
    abort;
end;
if(abs(f(a)) < eps) then
    error('solution at a')
    abort;
end;
if(abs(f(b)) < eps) then
    error('solution at b')
    abort;
end;
while (N > 0)
    c = (a+b)/2;
    if(abs(f(c)) < eps) then
        x = c;
        x
        return;
    end;
    if(f(a)*f(c) < 0) then
        b = c;
    else
        a = c;
    end;
N = N - 1;
end;
error('No convergence')
abort;
//end function
endfunction


function [x]=newton(x0,f,fp)
//newton-raphson algorithm
```

```
N = 100; eps = 1.e-4; // define max. no. iterations and error
maxval = 10000.0; // define value for divergence
xx = x0;
while (N>0)
    xn = xx-f(xx)/fp(xx);
    if(abs(f(xn))<eps) then
        x=xn
        disp(100-N);
        return(x);
    end;

    if (abs(f(xx))>maxval) then
        disp(100-N);
        error('Solution diverges');
        abort;
    end;
    N = N - 1;
    xx = xn;
end;
error('No convergence');
abort;
//end function
endfunction

deff('[y]=f(x)','y=x^3+4*x^2-10');
disp(half(1,2,f))

deff('[y]=fp(x)','y=3*x^2+8*x');
disp(newton(1,f,fp))
```

./personal_answer_Homework/lecture8_exe5.sci

Lecture 10

Solving Differential Equations I

10.1 Lecture

```
// #1: first-order, linear ODEs with constant coefficients
function [pInt] = intpoly(p)
//This fucntion calculates the indefinite integral
//of polynomial p
c = coeff(p);
n = length(c)-1;
d = [1];
for j=1:n+1
    d = [d j];
end;
cc = [0 c];
cc = cc./d;

disp('Indefinite integral - Add integration constant');
printf(' \n');
pInt = poly(cc,varn(p),'coeff');
endfunction
//end function intpoly


// #2: Solutions of homogeneous linear equations
// of any order with constant coefficients
deff('[FF]=f(s)',['f1=s(1)*exp(-a*t0)*cos(wI*t0+s(2))-x0';... //
    define f(1) with s(1)=A0, s(2)=phi0 x-x_0
'f2 = -s(1)*exp(-a*t0)*(a*cos(wI*t0+s(2))+wI*sin(wI*t0+s(2)))-v0'
    ;... // define f(2) with v-v_0
'FF = [f1;f2]']) // define FF=[f1 f2]
w0 = 0.7071; a = 0.05; wI = 0.7053; x0 = 1.5; v0 = -5.0; t0 = 0;
s0 = [5;%pi/3]
s = fsolve(s0,f) //f(s)=[0 0]
A0 = s(1); phi1 = s(2);
deff('[xs]=x(t)','xs = A0.*exp(-a.*t).*cos(wI.*t+phi1)')
```

```
deff('[vs]=v(t)',...
'vs =-A0.*exp(-a.*t).*(a.*cos(wI.*t+phi1)+wI.*sin(wI.*t+phi1))')
deff('[acc]=aa(t)',...
'acc=A0.*exp(-a.*t).*(a^2.*cos(wI.*t+phi1)+...
2.*a.*wI.*sin(wI.*t+phi1)-wI^2.*cos(wI.*t+phi1))')

// plot x-t v-t a-t with t: 0 -> 30
tt = [0:0.1:80]; xx = x(tt); vv = v(tt); aaa = aa(tt);
scf(1)
// using plot2d \not\plot
plot2d([tt',tt',tt'],[xx',vv',aaa'],[2,3,4],'111',...
'position@velocity@acceleration',[0 -10 80 10])
// from (0,-10) to (80,10)
xtitle('Damped oscillatory motion', 't', 'x,v,a')

// v-x plot with t: 0 -> 90
tt = [0:0.1:90]; xx = x(tt); vv = v(tt); aaa = aa(tt);
scf(2);
clf(2);
//xset('window',1);
plot(xx',vv')
xtitle('v-vs-x phase portrait','x','v')

scf(3);
clf(3);
//xset('window',2);
plot(xx',aaa')
xtitle('a-vs-x phase portrait','x','a')

scf(4);
clf(4);
//xset('window',3);
plot(vv',aaa')
xtitle('a-vs-v phase portrait','v','a')
```

./lecture/lecture_10_solving_differential_equations_i.sce

10.2 Homework

```
//lam = poly(0,'lam')
//p = lam^3-4*lam^2-11*lam+30
disp('Excercise 1 (a)')
p1 = poly([5 4 0 1],'x','coeff')
disp('function')
disp(p1)
disp('roots')
disp(lamda1 = roots(p1))

disp('Excercise 1 (b)')
p2 = poly([1 2 2],'x','coeff')
disp('function')
disp(p2)
disp('roots')
disp(lamda2 = roots(p2))

```

```
disp('Excercise 1 (c)')
p3 = poly([3 1 1 0 1],'x','coeff')
disp('function')
disp(p3)
disp('roots')
disp(lamda3 = roots(p3))

disp('Excercise 1 (d)')
p4 = poly([-3 0 1],'x','coeff')
disp('function')
disp(p4)
disp('roots')
disp(lamda4 = roots(p4))
```

./personal_answer_Homework/lecture10_exe1.sci

```
// 2a
disp("a")
deff('[FF]=f(s)',['f1=s(1)*exp(-a*t0)*cos(wI*t0+s(2))-x0';...
'f2 = -s(1)*exp(-a*t0)*(a*cos(wI*t0+s(2))+wI*sin(wI*t0+s(2)))-v0'
    ;...
'FF = [f1;f2]'])
m=2;b=0.01;k=2;
w0 = (k/m)^(1/2); a = b/(2*m); wI = (w0^2-a^2)^(1/2);
//w0 = 1; a = 0.0025; wI = 0.999997;
x0 = 0.2; v0 = 1.2; t0 = 0;
s0 = [5;%pi/3]
result=fsolve(s0,f)
A0 = result(1); phi1 = result(2);
//A0 = 7.1421364; phi1 = 1.3591997;
deff('[xs]=x(t)','xs = A0.*exp(-a.*t).*cos(wI.*t+phi1)')
deff('[vs]=v(t)',...
'vs =-A0.*exp(-a.*t).*(a.*cos(wI.*t+phi1)+wI.*sin(wI.*t+phi1))')
deff('[acc]=aa(t)',...
'acc=A0.*exp(-a.*t).*(a^2.*cos(wI.*t+phi1)+...
2.*a.*wI.*sin(wI.*t+phi1)-wI^2.*cos(wI.*t+phi1))')
// Plotting
tt = [0:0.1:30];
xx = x(tt);
vv = v(tt);
aaa = aa(tt);
xset('window',1);plot2d([tt',tt',tt'],[xx',vv',aaa'],[2,3,4],'111'
    ,...
'position@velocity@acceleration',[0 -10 30 10])
xtitle('Damped oscillatory motion', 't', 'x,v,a')

xset('window',2);plot(xx',vv')
xtitle('v-vs-x phase portrait','x','v')

xset('window',3);plot(xx',aaa')
xtitle('a-vs-x phase portrait','x','a')

xset('window',4);plot(vv',aaa')
xtitle('a-vs-v phase portrait','v','a')

// 2b
disp("b")
```

```
m=4; b=0.1; k=2;
w0 = (k/m)^(1/2); a = b/(2*m); wI = (w0^2-a^2)^(1/2);
s0 = [5;%pi/3]
result = fsolve(s0,f)
A0 = result(1); phi1 = result(2);
// Plotting
tt = [0:0.1:30];
xx = x(tt);
vv = v(tt);
aaa = aa(tt);
xset('window',5);plot2d([tt',tt',tt'],[xx',vv',aaa'],[2,3,4],'111'
    ,...
'position@velocity@acceleration',[0 -10 30 10])
xtitle('Damped oscillatory motion', 't', 'x,v,a')

xset('window',6);plot(xx',vv')
xtitle('v-vs-x phase portrait','x','v')

xset('window',7);plot(xx',aaa')
xtitle('a-vs-x phase portrait','x','a')

xset('window',8);plot(vv',aaa')
xtitle('a-vs-v phase portrait','v','a')

//2c
m=1; b=0.02; k=2;
w0 = (k/m)^(1/2); a = b/(2*m); wI = (w0^2-a^2)^(1/2);
s0 = [5;%pi/3]
result = fsolve(s0,f)
A0 = result(1); phi1 = result(2);
// Plotting
tt = [0:0.1:30];
xx = x(tt);
vv = v(tt);
aaa = aa(tt);
xset('window',9);plot2d([tt',tt',tt'],[xx',vv',aaa'],[2,3,4],'111'
    ,...
'position@velocity@acceleration',[0 -10 30 10])
xtitle('Damped oscillatory motion', 't', 'x,v,a')

xset('window',10);plot(xx',vv')
xtitle('v-vs-x phase portrait','x','v')

xset('window',11);plot(xx',aaa')
xtitle('a-vs-x phase portrait','x','a')

xset('window',12);plot(vv',aaa')
xtitle('a-vs-v phase portrait','v','a')

//2d
disp("d")
m=0.5; b=0.25; k=2;
w0 = (k/m)^(1/2); a = b/(2*m); wI = (w0^2-a^2)^(1/2);
s0 = [5;%pi/3]
result = fsolve(s0,f)
A0 = result(1); phi1 = result(2);
// Plotting
```

```
tt = [0:0.1:30];
xx = x(tt);
vv = v(tt);
aaa = aa(tt);
xset('window',13);plot2d([tt',tt',tt'],[xx',vv',aaa'],[2,3,4],'111'
    ,...
'position@velocity@acceleration',[0 -10 30 10])
xtitle('Damped oscillatory motion', 't', 'x,v,a')

xset('window',14);plot(xx',vv')
xtitle('v-vs-x phase portrait','x','v')

xset('window',15);plot(xx',aaa')
xtitle('a-vs-x phase portrait','x','a')

xset('window',16);plot(vv',aaa')
xtitle('a-vs-v phase portrait','v','a')
```

./personal_answer_Homework/lecture10_exe2.sci

```
//3a
function y=fa(x)
    y=sin(2*x)
endfunction
function y=dfa(x)
    y=2*cos(2*x)
endfunction
h=[1e-1 1e-2 1e-3 1e-4 1e5];
xx1=dfa(%pi);
error1=abs(xx1 - ((fa(%pi + h)-fa(%pi)) ./ h) );
xset('window',1);
clf(1);
plot2d(h,error1,1,'011',' ',[0 0 0.01 0.0002]);
plot2d(h,error1,-9,'011',' ',[0 0 0.01 0.0002]);
xtitle(['(a) error vs.' '$x$' '-increment'],'h','error');

clear;

//3b
function y=fb(x)
    y=(x.^2+3*x)./(x+1)
endfunction
// be careful of the dot here
function y=dfb(x)
    y=( x.^2 +2*x +3 ) ./ (1 + x).^2
endfunction
h=[1e-1 1e-2 1e-3 1e-4 1e-5];
xx2=dfb(2);
//function y=errorb(h)
//    y = abs (xx2 - ((f2(2 + h)-f2(2)) ./ h) );
//endfunction
//error2 = errorb(h);
error2 = abs (xx2 - ((fb(2+h) - fb(2)) ./ h) );
xset('window',2);
clf(2);
plot2d(h,error2,1,'011',' ',[0 0 0.01 0.001]);
plot2d(h,error2,-9,'011',' ',[0 0 0.01 0.001]);
```

```
xtitle('(b) error vs. x-increment','h','error');

clear;

// 3c
function y=f3(x)
    y=1 ./ (1+x.^2)
    // be careful of the dot here as well
endfunction
function y=df3(x)
    y=-(2*x) ./(1 + x^2)^2
endfunction
h=[1e-1 1e-2 1e-3 1e-4 1e-5];
xx3=df3(-1);
error3= abs ( xx3 - ( (f3(-1 + h)-f3(-1)) ./ h ) );
xset('window',3);
plot2d(h,error3,1,'011',' ',[0 0 0.1 0.1]);
plot2d(h,error3,-9,'011',' ',[0 0 0.1 0.05]);
xtitle('(c) error vs. x-increment',"$h$",'error');

clear;

// 3d
function y=f4(x)
    y=tan(x)
endfunction
function y=df4(x)
    y=sec(x)^2
endfunction
h=[1e-1 1e-2 1e-3 1e-4 1e-5];
xx4=df4(%pi/4);
error4=abs(xx4 - ( (f4(%pi/4 + h)-f4(%pi/4)) ./ h) );
xset('window',4);
plot2d(h,error4,1,'011',' ',[0 0 0.1 0.3]);
plot2d(h,error4,-9,'011',' ',[0 0 0.1 0.3]);
xtitle('(d) error vs. x-increment','h','error')
```

./personal_answer_Homework/lecture10_exe3.sci

```
//Exercise 4
function [x,y] = Euler1(x0,y0,xn,Dx,g)
//Euler 1st order method solving ODE
// dy/dx = g(x,y), with initial
//conditions y=y0 at x = x0. The
//solution is obtained for x = [x0:Dx:xn]
//and returned in y
ymaxAllowed = 1e+100;
x = [x0:Dx:xn];
y = zeros(x);
n = length(y);
y(1) = y0;
for j = 1:n-1
    y(j+1) = y(j) + Dx*g(x(j),y(j));
    if y(j+1) > ymaxAllowed then
        disp('Euler 1 - WARNING: underflow or overflow');
        disp('Solution sought in the following range:');
        disp([x0 Dx xn]);
```

```
        disp('Solution evaluated in the following range:');
        disp([x0 Dx x(j)]);
        n = j;
        x = x(1,1:n);
        y = y(1,1:n);
        break;
    end;
end;
endfunction
//End function Euler1
deff('[Df]=g(x,y)','Df=y*sin(x)')
[x1,y1]=Euler1(0,1,6.5,0.25,g);
[x2,y2]=Euler1(0,1,6.5,0.1,g);
[x3,y3]=Euler1(0,1,6.5,0.05,g);
xx=[0:0.1:2.5];
yy=exp(-cos(xx)+1);
ymin = min([y1 y2 y3 yy])
ymax = max([y1 y2 y3 yy])
rect = [0 0 7 25]
plot2d(x1,y1,-1,'011',' ',rect)
plot2d(x2,y2,-2,'011',' ',rect)
plot2d(x3,y3,-3,'011',' ',rect)
xtitle('Euler 1st order - dy/dx = y*sin(x)','x','y(x)')
```

./personal_answer_Homework/lecture10_exe4.sci

Lecture 11

Solving Differential Equations II

11.1 Lecture

```
//h = [1e-1 1e-2 1e-3 1e-4 1e-5 1e-6 1e-7 1e-8 1e-9];
//er = [0.00846132909 0.00098608109 0.0000]

function [x,y] = Euler1(x0,y0,xn,Dx,g)
//Euler 1st order method solving ODE
// dy/dx = g(x,y), with initial
//conditions y=y0 at x = x0. The
//solution is obtained for x = [x0:Dx:xn]
//and returned in y
ymaxAllowed = 1e+100;
x = [x0:Dx:xn];
y = zeros(x);
n = length(y);
y(1) = y0;
for j = 1:n-1
 y(j+1) = y(j) + Dx*g(x(j),y(j));
    if y(j+1) > ymaxAllowed then
          disp('Euler 1 - WARNING: underflow or overflow');
          disp('Solution sought in the following range:');
          disp([x0 Dx xn]);
          disp('Solution evaluated in the following range:');
          disp([x0 Dx x(j)]);  n = j; x = x(1,1:n); y = y(1,1:n);
          break;
    end;
end;
endfunction
//End function Euler1

//exec('Euler1.sci')
deff('[Df]=g(x,y)','Df = x+y')
[x1,y1]=Euler1(0,1,2,0.5,g);
[x2,y2]=Euler1(0,1,2,0.2,g);
```

```
[x3,y3]=Euler1(0,1,2,0.1,g);
[x4,y4]=Euler1(0,1,2,0.05,g);
xx = [0:0.1:2]; yy = -xx -1 + 2.* exp(xx);
scf(1);
clf(1);

plot2d(xx,yy,1,'011',' ',[0 0 2 12 ])
plot2d(x1,y1,-1,'011',' ',[0 0 2 12 ])
plot2d(x2,y2,-2,'011',' ',[0 0 2 12 ])
plot2d(x3,y3,-3,'011',' ',[0 0 2 12 ])
plot2d(x4,y4,-9,'011',' ',[0 0 2 12 ])

// Example 3
deff('[Df]=g(x,y)','Df=x+sin(x*y)')
[x1,y1]=Euler1(0,1,6.5,0.5,g);
[x2,y2]=Euler1(0,1,6.5,0.2,g);
[x3,y3]=Euler1(0,1,6.5,0.1,g);
[x4,y4]=Euler1(0,1,6.5,0.05,g);
//xx = [0:0.1:6.5]; yy = -xx -1 + 2.* exp(xx);
//ymin = min([y1 y2 y3 y4 yy])
//ymax = max([y1 y2 y3 y4 yy])
rect = [0 0 7 25]
scf(2);
clf(2);
//plot2d(xx,yy,1,'011',' ',rect)
plot2d(x1,y1,-1,'011',' ',rect)
plot2d(x2,y2,-2,'011',' ',rect)
plot2d(x3,y3,-3,'011',' ',rect)
plot2d(x4,y4,-9,'011',' ',rect)
xtitle('Euler 1st order - dy/dx = x+sin(x*y)','x','y(x)')

//Implicit Method Solving 1st order ODE
```

./lecture/lecture_11_solving_differential_equations_ii.sce

Lecture 12

Fourier Series

12.1 Lecture

```
function [yy] = fseries(a0,aa,bb,xx,L)
    nn = length(aa); mm = length(xx);yy = zeros(1,mm);
    for j = 1:mm
        yy = a0/2;
        for k = 1:nn
            yy = yy + aa(k)*cos(2*k*%pi*x/L) + bb(k)*sin(2*k*%pi*x/
    L);
        end;
    end;
endfunction
//end function fseries

function [a0,a,b,y] = fourierseries(n,c0,L,x,f,tol)
    deff('[gg1]=g1(xi)','gg1=f(xi)*cos(2*nn*%pi*xi/L)');
    deff('[gg2]=g2(xi)','gg2=f(xi)*sin(2*nn*%pi*xi/L)');
    deff('[aaa]=a1(nn)','aaa=(2/L)*intg(c0,c0+L,g1,tol)');
    deff('[bbb]=b1(nn)','bbb=(2/L)*intg(c0,c0+L,g2,tol)');
    a0 = (2/L)*intg(c0,c0+L,f,tol);
    nmax = max(n);a = []; b = [];
    for j = 1:nmax
        a = [a a1(j)]; b = [b b1(j)];
    end;
    m = length(n); k = length(x);y = zeros(m,k);
    for j =1:m
        aj = a(1:n(j)); bj = b(1:n(j));   y(j,:) = fseries(a0,aj,bj
    ,x,L);
    end;
endfunction
//end function fourierseries
```

./lecture/lecture_12_fourier.sce

12.2 Homework

```
function [yy] = fseries(a0,aa,bb,xx,L)
    nn = length(aa); mm = length(xx);yy = zeros(1,mm);
    for j = 1:mm
        yy = a0/2;
        for k = 1:nn
            yy = yy + aa(k)*cos(2*k*%pi*x/L) + bb(k)*sin(2*k*%pi*x/
    L);
        end;
    end;
endfunction
//end function fseries

function [a0,a,b,y] = fourierseries(n,c0,L,x,f,tol)
    deff('[gg1]=g1(xi)','gg1=f(xi)*cos(2*nn*%pi*xi/L)');
    deff('[gg2]=g2(xi)','gg2=f(xi)*sin(2*nn*%pi*xi/L)');
    deff('[aaa]=a1(nn)','aaa=(2/L)*intg(c0,c0+L,g1,tol)');
    deff('[bbb]=b1(nn)','bbb=(2/L)*intg(c0,c0+L,g2,tol)');
    a0 = (2/L)*intg(c0,c0+L,f,tol);
    nmax = max(n);a = []; b = [];
    for j = 1:nmax
        a = [a a1(j)]; b = [b b1(j)];
    end;
    m = length(n); k = length(x);y = zeros(m,k);
    for j =1:m
        aj = a(1:n(j)); bj = b(1:n(j));   y(j,:) = fseries(a0,aj,bj
    ,x,L);
    end;
endfunction
//end function fourierseries

deff('[y]=f(x)','y=exp(x)');
//exec('C:\Users\prgee\OneDrive\Year 2 Semester B\AP3114\Homeworks\
    Homework 10\fourierseries.sci')

L = 2; x=[-1:0.01:1]; y = f(x);
[a0,a,b,yy]=fourierseries([5,10,20,40],-L/2,L,x,f,1e-5);
scf(1);
clf(1);
plot2d([x' x' x' x' x'],[y' yy(1,:)' yy(2,:)' yy(3,:)' yy(4,:)']);
xtitle(['Fourier series for' '$y =e^{x}$' 'with 5,10,20,40
    components'],'x','y')
```

./personal_answer_Homework/lecture12_exe.sci

Appendix A

Mid-Term Exam

```
//Question 1a
disp("Question 1a")
real1=1
disp(real1)
disp("this is real variable");
string1="foo";
disp(string1)
disp("this is string");
boolean=%T;
disp(boolean)
disp("This is Booleans");
comple=%i;
disp(comple)
disp("This is complex number");
const=%pi;
disp(const)
disp("This is constant");

//Question 1b
disp("Question 1b")
Matrix=zeros(4,4)
for i=1:4
Matrix(i,i)=2
end
for i=1:3
Matrix(i,i+1)=-1
end
for i=1:3
Matrix(i+1,i)=1
end
disp(Matrix)


//Question 1c
disp("Question 1c")
A=linspace(1,10,10);
disp("A");
disp(A);
```

```
B=[2,3,4];
disp("B");
disp(B);
disp("C");
C=ones(1,3);
disp(C);


//Question 1d
disp("Question 1d")
Matrix2=zeros(4,4);
for i=1:4
    for j=1:4
        Matrix2(j,i)=j+i;
    end
end
disp(Matrix2)



//Question 2
disp("Question 2")
A=[1 2 3; 4 5 6];
B=[7 8 9; 10 11 12];
X1=A/B;
disp("X1");
disp(X1);
X2=A\B;
disp("X2");
disp(X2);


//Question 3
disp("Question 3")
BankB=100;
BankC=100;

BankA=100*(1+0.18);
disp("Bank A");
disp(BankA);

for i=1:12
    BankB=BankB*(1+0.015);
end
disp("Bank B");
disp(BankB);

for i=1:365
    BankC = BankC *(1+ 1.5 / 36524.25);
end
disp("Bank C");
disp(BankC);

disp("Bank B offers the best deal.");


//Question 4
```

```
disp("Question 4")
pi=0;
term=0;
while abs(%pi-pi) > 0.00005
    pi = pi + 4 * (((-1)^term ) /(2 * term + 1));
    term = term + 1;
end
disp("pi");
disp(pi);
disp("term");
disp(term);

//Question 5
disp("Question 5")
count=0;
for i=1:5000
   if rand()>0.5 then
       count=count+1;
   else
       count=count;
   end
   N(1,i)=count/i
end
for i=1:5000
    M(1,i)=0.5
end
//Plotting
figure(1);
scf(1);
clf(1);
xdata=linspace(1,5000,5000);
plot(xdata,N,"k")
plot(xdata,M,"--r")
xtitle("Sample Probability of Heads in n flips of a simulated coin"
    ,"Number of coin flips","Probability of heads")
legend("Sample Probability","Fair coin")


//Question 6a
disp("Question 6a")
t=linspace(0,2*%pi,1000);
figure(2);
scf(2);
clf(2);
plot(0.5*cos(t),0.5*sin(t),"pb")
plot(1*cos(t),1*sin(t),"pr")
plot(1.5*cos(t),1.5*sin(t),"pg")
plot(2*cos(t),2*sin(t),"py")
plot(2.5*cos(t),2.5*sin(t),"pk")
e=get("current_axes");
e.data_bounds=[-4,-2.5;4,2.5];


//Question 6b
disp("Question 6b")
figure(3);
scf(3);
```

```
clf(3);
plot(-1+0.6*cos(t),0.6*sin(t),"pb")
plot(0.6*cos(t),0.6*sin(t),"pk")
plot(1+0.6*cos(t),0.6*sin(t),"pr")
plot(-0.5+0.6*cos(t),-0.6+0.6*sin(t),"py")
plot(0.5+0.6*cos(t),-0.6+0.6*sin(t),"pg")
f=get("current_axes");
f.data_bounds=[-1.6,-1.5;1.6,1];
```

personal_answer_midterm.sci

Appendix B

Final Exam

```
//Problem 1
disp('Problem 1')
x = -2.1 : 0.15 : 2.1;
y = -6: 0.15 : 6;
[X, Y]= meshgrid(x, y);
U = 80 .* (Y.^2) .* (%e).^(- X.^2 - 0.3 * Y.^2)
// Plot
scf(1);
clf(1);
subplot(2,1,1);

surf(X,Y,U,'facecol','blu','edgecol','black');
xtitle(["$u(x,y)=80 y^2 \exp{-x^2 -0.3 y^2} $" 'U = 80 .* (Y.^2) .*
     (%e).^(- X.^2 - 0.3 * Y.^2)'])

subplot(2,1,2);
surf(X,Y,U);
xtitle(['$u(x,y)=80 y^2 \exp{-x^2 -0.3 y^2}$' 'U = 80 .* (Y.^2) .*
    (%e).^(- X.^2 - 0.3 * Y.^2)'])



// Problem 2
disp('Problem 2')
M1=[]
M2=[]
M3=[]
M4=[]
M5=[]
M6=[]

for n=1:100
    R1 = grand(1,2,"nor", 10, 2 )
    M1 = [M1 mean(R1)]

    R2 = grand(1,5,"nor", 10, 2 )
    M2 = [M2 mean(R2)]
```

```
    R3 = grand(1,10,"nor", 10, 2 )
    M3 = [M3 mean(R3)]

    R4 = grand(1,20,"nor", 10, 2 )
    M4 = [M4 mean(R4)]

    R5 = grand(1,50,"nor", 10, 2 )
    M5 = [M5 mean(R5)]

    R6 = grand(1,100,"nor", 10, 2 )
    M6 = [M6 mean(R6)]
end


scf(2);
clf(2);
subplot(2,3,1)
histplot(10,M1); //10 classes(blocks)
xtitle('n=2');
subplot(2,3,2)
histplot(10,M2); //10 classes(blocks)
xtitle('n=5');
subplot(2,3,3)
histplot(10,M3); //10 classes(blocks)
xtitle('n=10');
subplot(2,3,4)
histplot(10,M4); //10 classes(blocks)
xtitle('n=20');
subplot(2,3,5)
histplot(10,M5); //10 classes(blocks)
xtitle('n=50');
subplot(2,3,6)
histplot(10,M6); //10 classes(blocks)
xtitle('n=100');

S(1) = stdev (M1);
S(2) = stdev (M2);
S(3) = stdev (M3);
S(4) = stdev (M4);
S(5) = stdev (M5);
S(6) = stdev (M6);

scf(3);
clf(3);
plot([2 5 10 20 50 100],S','o-")
xtitle('uncertainty of the mean - sample size','sample size','
    uncertainty of the mean ')



// Problem 3
disp('Prblem 3')

function [x,y] = Euler1(x0,y0,xn,Dx,g)
//Euler 1st order method solving ODE
// dy/dx = g(x,y), with initial
//conditions y=y0 at x = x0. The
```

```
//solution is obtained for x = [x0:Dx:xn]
//and returned in y
ymaxAllowed = 1e+100;
x = [x0:Dx:xn];
y = zeros(x);
n = length(y);
y(1) = y0;
for j = 1:n-1
 y(j+1) = y(j) + Dx*g(x(j),y(j));
    if y(j+1) > ymaxAllowed then
          disp('Euler 1 - WARNING: underflow or overflow');
          disp('Solution sought in the following range:');
          disp([x0 Dx xn]);
          disp('Solution evaluated in the following range:');
          disp([x0 Dx x(j)]);  n = j; x = x(1,1:n); y = y(1,1:n);
          break;
    end;
end;
endfunction
//End function Euler1
deff('[Df]=g(x,y)','Df = 2 * x * y')
[x1,y1]=Euler1(0,1,2,0.5,g);
[x2,y2]=Euler1(0,1,2,0.1,g);
[x3,y3]=Euler1(0,1,2,0.05,g);
[x4,y4]=Euler1(0,1,2,0.01,g);
xx = [0:0.1:2]; yy = exp(xx.^2);
scf(4);
clf(4);

plot2d(xx,yy,1,'011',' ',[0 0 2 12 ])
plot2d(x1,y1,-1,'011',' ',[0 0 2 12 ])
plot2d(x2,y2,-2,'011',' ',[0 0 2 12 ])
plot2d(x3,y3,-3,'011',' ',[0 0 2 12 ])
plot2d(x4,y4,-9,'011',' ',[0 0 2 12 ])
xtitle(['Numerical solution of the equation'])

deff('[y]=f(x)','y = %e ^(x^2)')
// Integration
function y = Integral (dx)
xvalue = 0;
integral = 0;
while xvalue < 2;
    integral = integral + f(xvalue) * dx
    xvalue = xvalue + dx
end
y = integral;
endfunction

disp('Value of integral')
disp('dx = 0.5')
disp(Integral(0.5));
disp('dx = 0.1')
disp(Integral(0.1));
disp('dx = 0.05')
disp(Integral(0.05));
disp('dx = 0.01')
disp(Integral(0.01));
```

```
151
153 // Problem 4
155 disp("4(1)")
    mprintf ('Newton-Raphson method has a faster rate of convergence, \
        n but sometimes applying this method \n on some functions
        cannot converge to the root,\n it depends on initial point that
         we choose, \n another disadvantage is that the derivative of
        the function must be known.\n')
157 mprintf ('Bisection method has a relatively slow rate of
        convergence \n, but the root can always be found given the
        right interval.')
159
    disp('4(2)')
161 disp('(i)')
    function [x]=half(a,b,f)
163 //interval halving routine
    N = 100; eps = 1.e-2; // define max. no. iterations and error
165 if (f(a)*f(b) > 0) then
        error('no root possible f(a)*f(b) > 0')
167     abort;
    end;
169
    if(abs(f(a)) < eps) then
171     error('solution at a')
        abort;
173 end;
175 if(abs(f(b)) < eps) then
        error('solution at b')
177     abort;
    end;
179
    while (N > 0)
181     c = (a+b)/2;
        if (abs(f(c)) < eps) then
183         x = c;
            x
185         return;
        end;
187
        if (f(a)*f(c) < 0) then
189         b = c;
191     else
            a = c;
193     end;
        N = N - 1;
195 end;
    error('No convergence')
197 abort;
    //end function
199 endfunction
```

```
deff('[y] = f4 (x) ','y=x^2-10')
mprintf('Bisction method gives the root')
disp(half(1,5,f4));

disp('(ii)')
function [x]=newton(x0,f,fp)
//newton-raphson algorithm
N = 100; eps = 1.e-2; // define max. no. iterations and error
maxval = 10000.0; // define value for divergence
xx = x0;
while (N>0)
    xn = xx - f(xx)/fp(xx);
    if(abs(f(xn))<eps) then
        x = xn
//        disp(100-N);
        return(x);
    end;
    if (abs(f(xx))>maxval) then
//        disp(100-N);
        error('Solution diverges');
        abort;
    end;
    N = N - 1;
    xx = xn;
end;
error('No convergence');
abort;
//end function
endfunction
deff('[y] = df4 (x) ','y=2 * x')
mprintf('Newton-Raphson method gives the root')
disp(newton(1,f4,df4))

disp('(iii)')
mprintf(' In this case, yes.\n')
mprintf(' In some cases, if not, the initial point of Newton method
     should be changed \n to a more accurate value to obtain the
    root of this function. \n And the interval of bisection method
    should guaruntee that there is one and only one root in that
    interval.\n')

// Problem 5
disp('Problem 5')

height = 0;
volumbe = 0;
h = [0];
v = [0];
while h < 3
    height = height + 0.01;
    volumbe = (%pi /3) * height ^2 *(3 * 3 -height )
    h = [h height]
    v = [v volumbe]
end

while h < 7
    height = height + 0.01;
```

```
    volumbe = (2 * %pi /3) * 3^3 + %pi * 3^2 * (height - 3)
255     h = [h height]
    v = [v volumbe]
257 end


259
while h < 10
261     height = height + 0.01;
    volumbe = (4 * %pi /3) * 3^3 + %pi * 3^2 * (10-2*3) - (%pi /3 )
    *(10-height)^2*(3*3 -10 +height)
263     h = [h height]
    v = [v volumbe]
265 end

267 scf(5);
clf(5);
269 plot(h,v,'-')
xtitle('Volume of the Liduid',['h' '$(m)$'],['Volume' '$(m^3)$'])
```

personal_answer_final.sce