CE305 Numerical Methods for Civil Engineers **2012 Fall**

Solutions of Exercise v4.0

RULES

1. This is the **version 4.0**. In case there are any corrections for the solutions of Exercise 4, we will post an updated version on our website. You can follow the changes in the exercises by the **Version History** section below.

Version History

V4.0 Solutions of Exercise 4 are released.

1.

```
clear all
clc
%% Part a
A=zeros(20,40);
A(1:10,:)=1;
A(11:20,:)=2;
%% Part b
B=A;
for j=1:40
    B(11,j)=1/j;
end
%% Part c
C=B;
C(:,41)=3;
%% Part d
P=C;
for i=1:10
    P(i,i)=2*C(i,i);
end
%% Part e
Q=P;
Q(1,2)=7;
% Part f
R=Q.*Q;
% Part g
bigsum=0;
for i=1:20
    for j=1:41
        bigsum=bigsum+R(i,j);
    end
end
```

2.

```
clear all
close all
clc

U=[3 5 0 0 2 1
   0 9 0 0 0 6
   0 0 5 0 0 0
   9 8 4 5 2 6
   0 0 0 0 0 0
   3 5 0 0 0 0];

count=0; % to determine the size of the vector
for i=1:6 %searches the ith row
    for j=1:6 %searches the jth column
        if U(i,j)~=0
            count=count+1; %for each nonzero value, the size of the vector
increases one
            F(count)=U(i,j);
        end
    end
end
```

3.

```
clear all
clc
delta1=0; %initial guess
delta2=0; %initial guess
X=[delta1; delta2]; %vector of unknowns

for i=1:5
F1(i)=delta1(i)^2 + delta2(i)^2 + 6*delta2(i) - 0.1809;
F2(i)=delta1(i)^2 + delta2(i)^2 + 8*delta1(i) - 0.128;
F=[F1(i);F2(i)]; %formulate F vector
J11(i)=2*delta1(i);
J12(i)=2*delta2(i)+6;
J21(i)=2*delta1(i)+8;
J22(i)=2*delta2(i);
J=[J11(i) J12(i); J21(i) J22(i)]; %formulate Jacobian
X=X-inv(J)*F; % calculate the displacements
delta1(i+1)=X(1,1); % assign the new delta value which is calculated above, for
the new iteration
delta2(i+1)=X(2,1);
end
```

4.

a)

```
clear all
clc
count=1;
x=rand;
while x<0.8 || x>0.85
    count=count+1;
    x=rand;
end
disp(['It took' int2str(count) 'numbers this time'])
```

b)

```
clear all
clc
total=0;
count=0;
while total<20
    count=count+1;
    total=total+rand;
end
disp(['It took' int2str(count) 'numbers this time'])
```

c)

```
clear all
clc
count=0;
avg=0;
while abs(avg-0.5)>0.01
    count=count+1;
    avg=((count-1)*avg+rand)/count;
end
disp(['It took' int2str(count) 'numbers this time'])
```

5.

a)

```
clc
clear all
close all
x=0:8:16;
y=3+0.5*x+rand(1,size(x,2));
coun=0;
for i=0.0:0.1:max(x); % this is the x value at which you estimate
the polynomials value
    sum=0;
    for k=1:size(x,2)
        prod=1;
        for j=1:size(x,2)
            if k~=j
                prod=prod*(i-x(j))/(x(k)-x(j));
            end
        end
        sum=sum+prod*y(k);
    end
    coun=coun+1;
    xnew(coun,1)=i;
    xnew(coun,2)=sum;
end
```

b)

```
plot(xnew(:,1),xnew(:,2),x,y,'o')
xlabel('x');
ylabel('y');
```

6.

```
function y=funct(x)
if x<0
    error('x is smaller than 0, function is not defined!')
elseif 0<=x && x<1
    y=x^3;
elseif 1<=x && x<2
    y=2-(x^2)/5;
elseif 2<=x
    y=x+x^2;
end
end
```

```
clear all
clc

sum=0;
a=0;
b=3;

h=0.05;
n=(b-a)/h;

x=0;

for i=1:n
   sum=sum+(funct(x)+funct(x+h))*h/2;
   x=x+h;
end
```