W2 NUMERICAL ANALYSIS 2019-2020 FALL

Grading:

Class exercises : %30

Homework exercises %20

Final exam (written in class exam) open book %50

Books:

Numerical Analysis with example programs. M. Turhan ÇOBAN

http://www.turhancoban.com/kitap/NA_with%20example%20problems.pdf

Numerical Methods for Engineers, Steven C.Chapra, Raymond P. Canale, Mc Graw Hill publication

Numerical Analysis, Richard L. Burden J. Douglas Faires, Thomson Brooks/Cole publications

ROOT FINDING f(x)=0 x=?

CLASS EXERCISES

Class exercises will be completed and graded in class

EX 1

Java version

A reference empty function Shell (this program will be given to you it is existed in SCO1 directory.

interface if_x

```
import static java.lang.Math.*;
import java.io.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
  // single function single independent variable
  // example f=x*x
  // includes full set of derivatives
  // Reference :"Generation of Finite Difference Formulas on Arbitrary Spaced Grids",
  // Bength Fornberg, Mathematics of Computation, Volume 51, Number 184, October 1988
  // pages 699-706
  //interface version
@FunctionalInterface
interface if_x
{  public double func(double x);
//first order derivative
default double dfunc(double x)
{double h=1.0e-3;
 int n=1;
 int M=10;
 return dfunc(x,n,M,h);
}
//second order derivative
default double dfunc2(double x)
{double h=1.0e-3;
 int n=2;
 int M=10;
 return dfunc(x,n,M,h);
}
default double dfunc(double x,int N,int Mi,double hi)
{// order of the maximum derivative
 // N order of derivative
 // M degree of difference formula
double c[][][];
double alpha[];
double h;
int M;
double a[]=new double[0];
h=0.01;
double x0=0;
M=20;
```

```
double alphai[]={0,1,-1,2,-2,3,-3,4,-4,5,-5,6,-6,7,-7,8,-8,9,-9,10,-10,11,-11,12,-12,13,-13,14,-14,15,
-15,16,-16,17,-17,18,-18,19,-19,20,-20,21,-21,22,-22,23,-23,24,-24,25,-25,26,-26,27,-27,28,-28,29,-29,30,-30,
31,-31,32,-32,33,-33,34,-34,35,-35,36,-36,37,-37,38,-38,39,-39,40,-40,41,-41,42,-42,43,-43,44,-44,45,-45,46,
-46,47,-47,48,-48,49,-49,50,-50,51,-51,52,-52,53,-53,54,-54,55,-55,56,-56,57,-57,58,-58,59,-59,60,-60,
-61,61,-62,62,-63,63,-64,64,-65,65,-66,66,-67,67,-68,68,-70,70,-71,71,-72,72,-73,73,-74,74,-75,75,
-76,76,-77,77,-78,78,-79,79,-80,80,-81,81,-82,82,-83,83,-84,84,-85,85,-86,86,-87,87,
-88,88,-89,89,-90,90,-91,91,-92,92,-93,93,-94,94,-95,95,-96,96,-97,97,-98,98,-99,99,-100,100};
alpha=alphai;
int N1=alpha.length-1;
// M   degree of highest derivative
// N+1 number of coefficients
double delta[][][]=new double[N1+1][N1+1][M+1];
double c1,c2,c3;
delta[0][0][0]=1.0;
c1=1.0;
for(int n=1;n<=N1;n++)
{  c2=1;
   for(int nu=0;nu<=(n-1);nu++)
   {c3=alpha[n]-alpha[nu];
    c2=c2*c3;
    if(n<=M) delta[n-1][nu][n]=0.0;
    for(int m=0;m<=Math.min(n,M);m++)
    {
       if(m==0)
       { delta[n][nu][m]=((alpha[n]-x0)*delta[n-1][nu][m])/c3;}
       else
       {delta[n][nu][m]=((alpha[n]-x0)*delta[n-1][nu][m]-m*delta[n-1][nu][m-1])/c3;}
    }//next m
   }//next nu
   for(int m=0;m<=Math.min(n,M);m++)
   {   if(m==0)
                {delta[n][n][m]=c1/c2*(-(alpha[n-1]-x0)*delta[n-1][n-1][m]);}
                else
                {delta[n][n][m]=c1/c2*(m*delta[n-1][n-1][m-1]-(alpha[n-1]-x0)*delta[n-1][n-1][m]);}
   }//next m
   c1=c2;
}//next n
c=delta;
  if(Mi<N) M=N;
  else M=Mi;
  h=hi;
  double deriv=0;
  double h1=1/h;
  double h2=1;
  for(int j=0;j<N;j++)
  {h2*=h1;}
  for(int i=0;i<c[0].length;i++)
  {   deriv+=c[M][i][N]*func(x+alpha[i]*h);}
  return deriv*h2;}
default double dfunc(double x,int N)
{int M=30;double h=0.05*N;return dfunc(x,N,M,h);}
}
```

Bisection method(java version)

```
 import java.io.*;
import javax.swing.*;
import java.awt.*;
class bisection
{
//Bisection root finding method f(x)=0 x=? given initial guess limits a and b
public static double bisection(if_x f,double a,double b)
  {double b1=1.1*b;
   double r=(a+b)/2.0;
   double eps=1.0e-6;
   int nmax=100;
   int i=0;
   while(Math.abs(f.func(r))>eps && i<nmax)
   {if(f.func(a)*f.func(r)<0) b=r;
    else a=r;
    r=(a+b)/2.0;;
    i++;
   }
   if(i>=nmax) r=bisection(f,a,b1);
   return r;
  }
public static void main (String args[]) throws java.io.IOException
```

```
{    double a,b;
     a=Double.parseDouble(JOptionPane.showInputDialog(" INPUT LOWER LİMİT OF SEARCH AREA a : "));
     b=Double.parseDouble(JOptionPane.showInputDialog(" INPUT UPPER LIMIT OF SEARCH AREA b : "));
     double r;
     if_x f1=x->x*x-2.0;
     r= bisection(f1,a,b);
     String s1=" ROOT  : "+r+"\nFUNCTION VALUE : "+f1.func(r);
     JOptionPane.showMessageDialog(null,s1);
     System.exit(0);
}
}
```

C++ version

```
 #include <stdio.h>
#include <iostream>
#include <math.h>
#include <iomanip>
using namespace std;
// interface if_x
  class if_x{
  public: virtual double func(double x)=0;
  public: double dfunc(double x)
  { double h=0.0001;
    double hh=1.0/h;
    double df=(3.0*func(x-4.0*h)-32.0*func(x-3.0*h)+168.0*func(x-2.0*h)-672.0*func(x-h)+672.0*func(x+h)-168.0*func(x+2.0*h)
  +32.0*func(x+3.0*h) - 3.0*func(x+4.0*h))/840.0*hh;
    return df;
  }};

double bisection(if_x &f,double a,double b)
  {double b1=1.1*b;
   double r=(a+b)/2.0;
   double eps=1.0e-6;
   int nmax=100;
   int i=0;
   while(abs(f.func(r))>eps && i<nmax)
   {if(f.func(a)*f.func(r)<0) b=r;
    else a=r;
    r=(a+b)/2.0;
    i++;
   }
   if(i>=nmax) r=bisection(f,a,b1);
   return r;
  }
int main()
{  double a,b,r;
   class f1:public if_x{public:double func(double x){ return x*x-2; };};
   f1 f;
   cout << " INPUT LOWER LIMIT OF SEARCH AREA a : ";
   cin >>a;
   cout << " INPUT UPPER LIMIT OF SEARCH AREA b : ";
   cin >>b;
   r=bisection(f,a,b);
   cout<<setprecision(20);
   cout<<"r="<<r;
   return 0;
}
```

Python version

interface(abstract method) if_x

```
from abc import ABC, abstractmethod
class if_x(ABC):

    @abstractmethod
    def func(self,x):
        pass;

    def dfunc(self,x,n):
        dx=0.001;
        if n==0: df=self.func(x);
        elif n==1: df=(3.0*self.func(x-4.0*dx)-32.0*self.func(x-3.0*dx)+168.0*self.func(x-2.0*dx)-672.0*self.func(x-
dx)+672.0*self.func(x+dx)-168.0*self.func(x+2.0*dx)+32.0*self.func(x+3.0*dx)-3.0*self.func(x+4.0*dx))/840.0/dx;
```

```
        elif n==2: df=(-14350.0*self.func(x)-9.0*self.func(x-4.0*dx)+128.0*self.func(x-3.0*dx)-1008.0*self.func(x-
2.0*dx)+8064.0*self.func(x-dx)+8064.0*self.func(x+dx)-1008.0*self.func(x+2.0*dx)+128.0*self.func(x+3.0*dx)-
9.0*self.func(x+4.0*dx))/5040.0/dx/dx;
        elif n==3: df=(-7.0*self.func(x-4.0*dx)+72.0*self.func(x-3.0*dx)-338.0*self.func(x-2.0*dx)+488.0*self.func(x-dx)-
488.0*self.func(x+dx)+338.0*self.func(x+2.0*dx)-72.0*self.func(x+3.0*dx)+7.0*self.func(x+4.0*dx))/240.0/dx/dx/dx;
        elif n==4: df=(2730.0*self.func(x)+7.0*self.func(x-4.0*dx)-96.0*self.func(x-3.0*dx)+676.0*self.func(x-2*dx)-1952.0*self.func(x-
dx)-1952.0*self.func(x+dx)+676.0*self.func(x+2.0*dx)-96.0*self.func(x+3.0*dx)+7.0*self.func(x+4.0*dx))/240.0/dx/dx/dx/dx;
        elif n==5: df=(self.func(x-4.0*dx)-9.0*self.func(x-3.0*dx)+26.0*self.func(x-2.0*dx)-29.0*self.func(x-dx)+29.0*self.func(x+dx)-
26.0*self.func(x+2.0*dx)+9.0*self.func(x+3.0*dx)-self.func(x+4.0*dx))/6.0/dx/dx/dx/dx/dx;
        elif n==6: df=(-150.0*self.func(x)-self.func(x-4.0*dx)+12.0*self.func(x-3.0*dx)-52.0*self.func(x-2.0*dx)+116.0*self.func(x-
dx)+116.0*self.func(x+dx)-52.0*self.func(x+2.0*dx)+12.0*self.func(x+3.0*dx)-self.func(x+4.0*dx))/4.0/dx/dx/dx/dx/dx/dx;
        elif n==7: df=(-self.func(x-4.0*dx)+6.0*self.func(x-3.0*dx)-14.0*self.func(x-2.0*dx)+14.0*self.func(x-dx)-
14.0*self.func(x+dx)+14.0*self.func(x+2.0*dx)-6.0*self.func(x+3.0*dx)+self.func(x+4.0*dx))/2.0/dx/dx/dx/dx/dx/dx/dx;
        elif n==8: df=(70.0*self.func(x)+self.func(x-4.0*dx)-8.0*self.func(x-3.0*dx)+28.0*self.func(x-2.0*dx)-56.0*self.func(x-dx)-
56.0*self.func(x+dx)+28.0*self.func(x+2.0*dx)-8.0*self.func(x+3.0*dx)+self.func(x+4.0*dx))/dx/dx/dx/dx/dx/dx/dx/dx;
        else: df=0;
        return df;
```

```
 import math;

def sin(x):
    total=0;
    factorial=1;
    power=x;
    plusminus=1;
    i=0;
    for k in range(1,80):
        total=total+power/factorial*plusminus;
        power=power*x*x;
        i=2*k+1;
        factorial=factorial*(i-1)*i;
        plusminus*=-1;
    return total;

pi=math.pi;
x=pi/3.0;
print ("sin="+str(sin(x))+"math.sin = "+str(math.sin(x)));
```

Octave(matlab) version

Octave editor (write the fuction sin1)

```
 function xr=bisection(f,xl,xu)
% bisection root finding method
maxit=100;
iter=0;
es=0.0000001;
ea=1.1*es;
while((ea>es)&&(iter<maxit))
    xr=(xl+xu)/2.0;
    iter=iter+1;
    if xr~=0 ea=abs((xu-xl)/(xu+xl))*100;end
    fxl= f(xl);
    fxr= f(xr);
    test= fxl*fxr;
    if test == 0.0      ea=0;
    elseif test < 0.0  xu=xr;
    else               xl=xr;
    end
end
if(iter>=maxit) fprintf('Maximum number of iteration is exceeded result might not be valid","MAKSİMUM NUMBER OF ITERATION
WARNING');
end
end
```

Octave command window

```
>> format long
>> f=@(x)x*x-2;
>> bisection(f,1,3)
ans =  1.414213561452925
```

EX2: Secant method Java version

```
// ROOT FINDING Secant method
import java.util.*;
import java.awt.*;
import java.applet.Applet;
import java.awt.event.*;
import javax.swing.*;

public class secant
{
public static double secant(if_x f,double x)
{ double eps=1.0e-10;
  double y=f.dfunc(x);
  double dy=f.dfunc2(x);
  int miter=100;
  int i=0;
  while(Math.abs(y)>eps && i<miter)
  {x=x-y/dy;
   y=f.func(x);
   dy=f.dfunc(x);
   i++;
  }
  if(i>=miter)
    System.out.println("i="+i+"results may not be valid");
  return x;
}
     public static void main (String args[])
     { if_x f=x->x*x-2;
     double x=Double.parseDouble(JOptionPane.showInputDialog(" enter lower limit of the function a : "));
     double r= secant(f,x);
     JOptionPane.showMessageDialog(null," optimisation value  : "+r+"\nFunction value : "+f.func(r),
     "secant optimisation : ",JOptionPane.PLAIN_MESSAGE);
     Plot p=new Plot(f,0,3);
     p.plot();
     }
}
```

Secant method C++ version

```
#include <stdio.h>
#include <iostream>
#include <math.h>
#include <iomanip>
using namespace std;

class if_x{
public: virtual double func(double x)=0;
public: double dfunc(double x)
{ double h=0.0001;
  double hh=1.0/h;
  double df=(3.0*func(x-4.0*h)-32.0*func(x-3.0*h)+168.0*func(x-2.0*h)-672.0*func(x-h)+672.0*func(x+h)-168.0*func(x+2.0*h)
+32.0*func(x+3.0*h) - 3.0*func(x+4.0*h))/840.0*hh;
  return df;
}
};

double secant(if_x &f,double x)
{
int nmax=100;
double tolerance=1.0e-8;
double fx,dfx;
for(int i=0;i<nmax;i++)
{ fx=f.func(x);
  dfx=f.dfunc(x);
  x-=fx/dfx;
  cout <<"i="<<i<<"x="<<x<<"fx="<<fx<<"dfx="<<dfx<<"\n";
  if(fabs(fx)<tolerance) { return x;}
}
cout <<("Maximum number of iteration is exceeded \n result might not be valid");
return x;
}
int main()
{  double x,r;
   class f1:public if_x{public:double func(double x){ return x*x-2; };};
   f1 f;
   cout << " input initial guess x : ";
```

```
   cin >>x;
   r=secant(f,x);
   cout<<setprecision(20);
   cout<<"r="<<r;
   return 0;
}
```

Secant method python version

```
from math import *
from if_x import *;

def secant(f,x):
   # secant root finding method
  nmax=100
  tolerance=1.0e-10
  for i in range(0,nmax):
      fx= f.func(x);
      dfx=f.dfunc(x,1);
      x=x-fx/dfx
      if abs(fx)<tolerance: return x
  return x

class f1(if_x):func=lambda self,x:x*x-2;
f=f1();
x=float(input("enter initial guess : "))
r=secant(f,x);
print(" ROOT  : ",r,"\nFUNCTION VALUE : ",f.func(r));
```

Secant method octave version

editor

```
function p=secant(f,x,tolerance,nmax)
% secant method for root finding
if nargin<4 nmax=100;end
if nargin<3 tolerance=1e-10;end
i=0;
fx=f(x);
while (abs(fx) > tolerance) && i<nmax
    fx=f(x);
    dfx=df(f,x);
    x=x-fx/dfx;
    i=i+1;
end
p=x;
if i>=nmax fprintf('Maximum number of iterations is exceeded the result may not be valid');end
end
```

Command window

```
>> format long
>> f=@(x)x*x-2;
>> secant(f,1)
ans =  1.414213562373095
>>
```

HOMEWORK EXERCISES

Homework exercises will be done at home and will bring to next Thursday class printed no late exercises will be excepted. Each code should include student name id#, code plus results should be given. Homeworks will be accepted in written format plus a computer copy in pdf format will be sent to numerical_analysis@turhancoban.com adress your file name should be "group"+"week#"+studentname+studentid#.pdf

A_W1_turhan_coban_0101333.pdf

B_W3_ali_veli_02335646.pdf

W1HW1 : function f(x) $= x * x - 5 * x + 1$ is given find the root by using

a) Newton raphson method by hand
b) Secant method by hand

c) Secant method by computer program (you can select your language)
d) Bisection method by computer program (you can select your language)

W1HW2 : function f(x) $= \sin(x)$ is given find the root in the range of $0.2 \leq x \leq 2.5$ by using

a) Newton raphson method by hand
b) Bisection method by hand
c) Secant method by computer program (you can select your language)
d) Bisection method by computer program (you can select your language)