W5 NUMERICAL ANALYSIS 2019-2020 FALL

OPTIMISATION I

CLASS EXERCISES
Class exercises will be completed and graded in class

EX 1 Following excel files(programs) are given investigate solutions

opt_golden_search.xlsx **opt_secant.xlsx** **opt_bisection.xlsx**	

EX2 following one dimensional nonlinear optimiation programs are given investigate solutions.

```
public class opt_test
{
public static double golden(if_x f,double a,double b)
{
//  find the minimum of the function
// note  maximum f(x) = minimum (-f(x))
   double epsilon;
   double delta;
   int print;
   epsilon=0.001;
   delta=0.002;
   print=0;
   double r1 = (Math.sqrt(5.0)-1.0)/2.0; // golden ratio
   double r2 = r1*r1;
   double h = b - a;
   double ya = f.func(a);
   double yb = f.func(b);
   double c = a + r2*h;
   double d = a + r1*h;
   double yc = f.func(c);
   double yd = f.func(d);
   int k = 1;
   double dp,dy,p,yp;
   while ((Math.abs(yb-ya)>epsilon) || (h>delta))
   {
    k++;
    if (yc<yd)
      {
      b = d;
      yb = yd;
      d = c;
      yd = yc;
      h = b - a;
      c = a + r2 * h;
      yc = f.func(c);
      }
    else
     {
       a = c;
       ya = yc;
       c = d;
       yc = yd;
       h = b - a;
       d = a + r1 * h;
       yd = f.func(d);
     }//end of if
  }//end of while
  dp = Math.abs(b-a);
  dy = Math.abs(yb-ya);
  p = a;
  yp = ya;
  if (yb<ya)
  {
   p = b;
```

```
      yp = yb;
    }
    if(print==1)
    {System.out.println("x min = "+p+"ymin = "+yp+"error of x ="+dp+"error of y"+dy); }
    return p;
}
public static double quadratic_polynomial(if_x f,double x0,double x2)
{double x1=(x0+x2)/2.0;
 return quadratic_polynomial(f,x0,x1,x2);
}
//secant optimisation method
public static double secant(if_x f,double x)
{ double eps=1.0e-10;
  double y=f.func(x);
  double dy=f.dfunc(x);
  int miter=100;
  int i=0;
  while(Math.abs(y)>eps && i<miter)
  {x=x-y/dy;
   y=f.dfunc(x);
   dy=f.dfunc2(x);
   i++;
  }
  if(i>=miter)
    System.out.println("i="+i+"results may not be valid");
  return x;
}
//Bisection optimisation 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.dfunc(r))>eps && i<nmax)
   {if(f.dfunc(a)*f.dfunc(r)<0) b=r;
    else a=r;
    r=(a+b)/2.0;
    System.out.println("r="+r+"i="+i);
    i++;
   }
   return r;
  }
public static double quadratic_polynomial(if_x f,double x0,double x1,double x2)
{
double epsilon=1.0e-10;
double delta=1.0e-5;
int print=0;
return quadratic_polynomial(f,x0,x1,x2,epsilon,delta,print);
}
public static double quadratic_polynomial(if_x f,double x0,double x1,double x2,double epsilon,double delta,int print)
{
int maxit=100;
double f0 = f.func(x0);
    double f1 = f.func(x1);
    double f2 = f.func(x2);
    double f3 = f2;
    double x3 = 0;
    double h = x1 - x0;
    double k=1;
    double dd=Math.abs(f1-f0);
    while ((dd >epsilon) || (h>delta))
    {
      k++;
      x3=(f0*(x1*x1-x2*x2)+f1*(x2*x2-x0*x0)+f2*(x0*x0-x1*x1))/
            (2*f0*(x1-x2)+2.0*f1*(x2-x0)+2.0*f2*(x0-x1));
      f3=f.func(x3);
      if(x3 >= x0 && x3<x1)
        {x2=x1;f2=f1;x1=x3;f1=f3;}
      else if(x3 >= x1 && x3 <x2)
        {x0=x1;f0=f1;x1=x3;f1=f3; }
      else if(x3 >x2)
        {x0=x1;f0=f1;x1=x2;f1=f2;x2=x3;f2=f3; }
      else if(x3 < x0)
        {x0=x3;f0=f3;x1=x0;f1=f0;x2=x1;f2=f1; }
```

```
      dd=Math.abs(f1-f0);
      h=Math.abs(x1-x0);
      if(k>maxit) break;
    }//end of while
    if(print==1)
    {System.out.println("x  = "+x3+"f = "+f3+"hata x ="+h+"hata f(x)=y "+dd); }
    return x3;
}
public static void main(String arg[])
{if_x ff=(x)->1.0/3.0*x*x*x-2*x+5;
 double r1=secant(ff,1.0);
 double r2=bisection(ff,1.0,3.0);
 double r3=golden(ff,1.0,3.0);
 double r4=quadratic_polynomial(ff,1.0,3.0);
 System.out.println("r1="+r1+"r2="+r2+"r3="+r3+"r4="+r4);
 Plot pp=new Plot(ff,0.0,3.0);
 pp.plot();
 if_x df=(x)->ff.dfunc(x);
 Plot pp1=new Plot(df,0.0,3.0);
 pp1.plot();
}
}
```

EX3 following multi dimensional nonlinear optimiation programs are given investigate solutions.

```
import java.util.Locale;
public class opt1_test
{
public static double[] gauss(double a[][],double b[])
  { //gauss elimination with partial pivoting
  int n=b.length;
  double x[]=new double[n];
  double carpan=0;
  double toplam=0;
  double buyuk;
  double dummy=0;
  //gauss elimination
  int i,j,k,p,ii,jj;
  for(k=0;k<(n-1);k++)
  { //pivoting
                p=k;
                buyuk=Math.abs(a[k][k]);
                for(ii=k+1;ii<n;ii++)
                {  dummy=Math.abs(a[ii][k]);
                  if(dummy > buyuk) {buyuk=dummy;p=ii;}
                }
                if(p!=k)
                { for(jj=k;jj<n;jj++)
                  { dummy=a[p][jj];
                    a[p][jj]=a[k][jj];
                    a[k][jj]=dummy;
                  }
                  dummy=b[p];
                  b[p]=b[k];
                  b[k]=dummy;
                }
                //Solving gauss elimination
    for(i=k+1;i<n;i++)
    {  carpan=a[i][k]/a[k][k];
       a[i][k]=0;
       for(j=k+1;j<n;j++)
       {   a[i][j]-=carpan*a[k][j]; }
          b[i]   =b[i]   -carpan*b[k];
                }
  }
  //back substitution
  x[n-1]=b[n-1]/a[n-1][n-1];
  for(i=n-2;i>=0;i--)
  {
    toplam=0;
    for(j=i+1;j<n;j++)
    {  toplam+=a[i][j]*x[j];}
    x[i]=(b[i]-toplam)/a[i][i];
```

```
  }
  return x;
  }
public static double[] newton_opt(if_xj f,double x[])
{ int k=x.length;
  int nmax=100;
  double tolerance=1.0e-10;
  double fx[];
  double dfx[][];
  double dx[];
  double total=0;
  for(int i=0;i<nmax;i++)
  { fx=f.dfunc(x);
    dfx=f.d2func(x);
    dx=gauss(dfx,fx);
    for(int j=0;j<k;j++)
      x[j]-=dx[j];
    total=0;
    for(int j=0;j<k;j++)
      total+=fx[j];
    if(Math.abs(total)<tolerance) return x;
  }
  return x;
}
public static String toString(double a[])
  {int n=a.length;
   String s="";
   for(int i=0;i<n;i++)
   {s+=String.format(Locale.US,"%10g",a[i])+" ";}
   return s;
  }
  public static String toString(double a[][])
  {int n=a.length;
   int m=a[0].length;
   String s="";
   for(int i=0;i<n;i++)
   { for(int j=0;j<m;j++)
                {s+=String.format(Locale.US,"%10g",a[i][j])+" ";}
                s+="\n";
   }
   return s;
  }
//6-2. Nelder and Mead Simplex multivariable nonlinear optimization method
//  Nelder & Mead 1965 Computer J, v.7, 308-313.
//_____________________________________________________________________
public static double[] nelder(if_xj fnelder,double a[],double da[],int maxiteration,double tolerance,int printlist)
{
            int i,j;
double x[][]=new double[a.length+1][a.length];
double p[][]=new double[a.length+1][a.length+1];
for(i=0;i<x.length;i++)
   {for(j=0;j<x[0].length;j++)
     {if(i==j){x[i][j]=a[j]+da[j];p[i][j]=x[i][j];}
      else    {x[i][j]=a[j];p[i][j]=x[i][j]; }
     }
     p[i][j] = fnelder.func(p[i]);
   }

   // Inlet variable definitions
   // fnelder : abstract multivariable function f(x)
   // x : independent variable set of n+1 simplex elements
   // maxiteration : maximum iteration number
   // tolerance :
        int NDIMS = x.length-1;
        int NPTS = x.length;
        int FUNC = NDIMS;
        int ncalls = 0;
          ////// construct the starting simplex ///////////////////
          //double p[][]=new double[NPTS][NPTS]; // [row][col] = [whichvx][coord,FUNC]
          double z[]=new double[NDIMS];
          double best = 1E99;
          ///////////////// calculate the first function values for the simplex ////////////////
          int iter=0;
          for (iter=1; iter<maxiteration; iter++)
          {
```

```
            //////////  define lo, nhi, hi (low high next_to_high //////////////
int  ilo=0, ihi=0, inhi = -1; // -1 means missing
double flo = p[0][FUNC];
double fhi = flo;
double pavg,sterr;
for (i=1; i<NPTS; i++)
{
  if (p[i][FUNC] < flo)
    {flo=p[i][FUNC]; ilo=i;}
  if (p[i][FUNC] > fhi)
    {fhi=p[i][FUNC]; ihi=i;}
}
double fnhi = flo;
inhi = ilo;
for (i=0; i<NPTS; i++)
 if ((i != ihi) && (p[i][FUNC] > fnhi))
   {fnhi=p[i][FUNC]; inhi=i;}
/////////// exit criteria //////////////
if ((iter % 4*NDIMS) == 0)
{
   // calculate the avarage (including maximum value)
   pavg=0;
   for(i=0;i<NPTS;i++)
     pavg+=p[i][FUNC];
   pavg/=NPTS;
   double tot=0;
   if(printlist!=0)
   {  System.out.print(iter);
      for (j=0; j<=NDIMS; j++)
         { System.out.print(p[ilo][j]+" ");}
      System.out.println("");
   }
   for(i=0;i<NPTS;i++)
   { tot=(p[i][FUNC]-pavg)*(p[i][FUNC]-pavg);}
   sterr=Math.sqrt(tot/NPTS);
   //if(sterr < tolerance)
     { for (j=0; j<NDIMS; j++)
          { z[j]=p[ilo][j];}
       //break;
     }
   best = p[ilo][FUNC];
}

///// calculate avarage without maximum value //////

double ave[] = new double[NDIMS];
for (j=0; j<NDIMS; j++)
 ave[j] = 0;
for (i=0; i<NPTS; i++)
 if (i != ihi)
   for (j=0; j<NDIMS; j++)
     ave[j] += p[i][j];
for (j=0; j<NDIMS; j++)
  ave[j] /= (NPTS-1);
//////// reflect ////////////////
double r[] = new double[NDIMS];
for (j=0; j<NDIMS; j++)
 r[j] = 2*ave[j] - p[ihi][j];
double fr = fnelder.func(r);

if ((flo <= fr) && (fr < fnhi))  // in zone: accept
{
   for (j=0; j<NDIMS; j++)
     p[ihi][j] = r[j];
   p[ihi][FUNC] = fr;
   continue;
}
if (fr < flo)  //// expand
{
   double e[] = new double[NDIMS];
   for (j=0; j<NDIMS; j++)
     e[j] = 3*ave[j] - 2*p[ihi][j];
   double fe = fnelder.func(e);
   if (fe < fr)
   {
```

```
                for (j=0; j<NDIMS; j++)
                  p[ihi][j] = e[j];
                p[ihi][FUNC] = fe;
                continue;
              }
              else
              {
                for (j=0; j<NDIMS; j++)
                  p[ihi][j] = r[j];
                p[ihi][FUNC] = fr;
                continue;
              }
            }
            ////////////// shrink:
            if (fr < fhi)
            {
              double c[] = new double[NDIMS];
              for (j=0; j<NDIMS; j++)
                c[j] = 1.5*ave[j] - 0.5*p[ihi][j];
              double fc = fnelder.func(c);
              if (fc <= fr)
              {
                for (j=0; j<NDIMS; j++)
                  p[ihi][j] = c[j];
                p[ihi][FUNC] = fc;
                continue;
              }
              else   /////// shrink
              {
                for (i=0; i<NPTS; i++)
                  if (i != ilo)
                  {
                    for (j=0; j<NDIMS; j++)
                      p[i][j] = 0.5*p[ilo][j] + 0.5*p[i][j];
                    p[i][FUNC] = fnelder.func(p[i]);
                  }
                continue;
              }
            }

            if (fr >= fhi)   ///
            {
              double cc[] = new double[NDIMS];
              for (j=0; j<NDIMS; j++)
                cc[j] = 0.5*ave[j] + 0.5*p[ihi][j];
              double fcc = fnelder.func(cc);
              if (fcc < fhi)
              {
                for (j=0; j<NDIMS; j++)
                  p[ihi][j] = cc[j];
                p[ihi][FUNC] = fcc;
                continue;
              }
              else    /////////
              {
                for (i=0; i<NPTS; i++)
                  if (i != ilo)
                  {
                    for (j=0; j<NDIMS; j++)
                      p[i][j] = 0.5*p[ilo][j] + 0.5*p[i][j];
                    p[i][FUNC] = fnelder.func(p[i]);
                  }
              }
            }
          }
      return z;
}
    public static double[] nelder(if_xj fnelder,double a[],double da[],double tolerance)
    {return nelder(fnelder,a,da,500,tolerance,0);}

    public static double[] nelder(if_xj fnelder,double a[],double da[])
    {return nelder(fnelder,a,da,500,1.0e-10,0);}

    public static double[] nelder(if_xj fnelder,double a[])
    {
```

```
                    double [] da=new double[a.length];
            for(int i=0;i<a.length;i++) da[i]=0.1*a[i];
            return nelder(fnelder,a,da);
      }


public static void main(String arg[])
{if_xj ff=(double x[])->3.0*x[0]*x[0]-4.0*x[0]*x[1]+2.0*x[1]*x[1]-x[0]-x[1];
 double xx[]={1,2};
 double r1[]=newton_opt(ff,xx);
 double r2[]=nelder(ff,xx);
 System.out.println("r1=\n"+toString(r2)+"\nr2=\n"+toString(r2));
 surfacePlot pp1=new surfacePlot(ff,-2.0,2.0,-2.0,2.0);
 pp1.plot();
}
}
```

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * Example.java
 *
 * Created on Mar 13, 2010, 2:46:54 PM
 */
/**
 *
 * @author siva
 */
//changed by M. Turhan Coban
import java.awt.*;
import javax.swing.*;
import javax.swing.JPanel;
import javax.swing.JLabel;
import java.awt.BorderLayout;
import java.awt.FlowLayout;

public class surfacePlot2 extends javax.swing.JFrame {
                ExampleSurfaceModel2 model;
                SurfaceCanvas canvas;
                JPanel southPanel;
                Container c;
      public void plot()
      {
                c=this.getContentPane();
      c.setLayout(new BorderLayout());
      southPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 50, 5));
      southPanel.add(new JLabel("Rotate: Mouse Click & Drag"));
      southPanel.add(new JLabel("Zoom: Shift Key + Mouse Click & Drag"));
      southPanel.add(new JLabel("Move: Control Key + Mouse Click & Drag"));
      add(southPanel, BorderLayout.SOUTH);
                this.setSize(1000,1000);
                add(canvas, BorderLayout.CENTER);
                this.repaint();
                this.setVisible(true);
   }

   public surfacePlot2(if_xj f,double x0,double xn,double y0,double yN,double z0,double zN,
            String xLabel, String yLabel,String zLabel)
   {   model = new ExampleSurfaceModel2(f,x0,xn,y0,yN,z0,zN,xLabel,yLabel,zLabel);
                canvas = new SurfaceCanvas();
                canvas.setModel(model);
                canvas.setSize(1000,1000);
                canvas.setVisible(true);

            }
   public surfacePlot2(if_xj f,double x0,double xN,double y0,double yN,double z0,double zN)
   {   String xLabel="x";
      String yLabel="y";
      String zLabel="z";
                model = new ExampleSurfaceModel2(f,x0,xN,y0,yN,z0,zN,xLabel,yLabel,zLabel);
                canvas = new SurfaceCanvas();
                canvas.setModel(model);
```

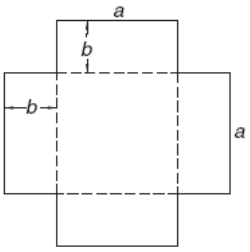
```
                canvas.setSize(1000,1000);
                canvas.setVisible(true);
            }
            public surfacePlot2(if_xj f,double x0,double xN,double y0,double yN)
    {   String xLabel="x";
        String yLabel="y";
        String zLabel="z";
        double z0=9.9e50;
        double zN=-9.9e50;
        int n=100;
        double dx=(xN-x0)/n;
        double dy=(yN-y0)/n;
        double zz;
        double x1=0,y1=0;
        for(int i=0;i<n;i++)
        {for(int j=0;j<n;j++)
         {   x1=x0+i*dx;y1=y0+j*dy;
            double xx[]={x1,y1};
                    zz=f.func(xx);
                    if(zz<z0) z0=zz;
                    if(zz>zN) zN=zz;
        }
        }
                model = new ExampleSurfaceModel2(f,x0,xN,y0,yN,z0,zN,xLabel,yLabel,zLabel);
                canvas = new SurfaceCanvas();
                canvas.setModel(model);
                canvas.setSize(1000,1000);
                canvas.setVisible(true);
            }
            public surfacePlot2(if_xj f,double x0,double xN,double y0,double yN,String xLabel,String yLabel,String zLabel,int N)
    {
        double z0=9.9e50;
        double zN=-9.9e50;
        int n=100;
        double dx=(xN-x0)/n;
        double dy=(yN-y0)/n;
        double zz;
        double x1=0,y1=0;
        for(int i=0;i<n;i++)
        {for(int j=0;j<n;j++)
         {   x1=x0+i*dx;y1=y0+j*dy;
            double xx[]={x1,y1};
                    zz=f.func(xx);
                    if(zz<z0) z0=zz;
                    if(zz>zN) zN=zz;
        }
        }
                model = new ExampleSurfaceModel2(f,x0,xN,y0,yN,z0,zN,xLabel,yLabel,zLabel);
                canvas = new SurfaceCanvas();
                canvas.setModel(model);
                canvas.setSize(1000,1000);
                canvas.setVisible(true);
            }
    public static void main(String args[]) {
                if_xj f=(double x[])->x[0]*x[0]+x[1]*x[1];
                surfacePlot2 p=new surfacePlot2(f,-5.0,5.0,-5.0,5.0);
        p.plot();
    }
}
```

```
/*
* To change this template, choose Tools | Templates
* and open the template in the editor.
*/
//package net.sf.surfaceplot;
/**
 * @author salagarsamy
*/
// Adapted by M. Turhan Coban
public class ExampleSurfaceModel2 implements ISurfacePlotModel
{   float x[],y[],z[];
    String xlabel,ylabel,zlabel;
    float xmin,xmax,ymin,ymax,zmin,zmax;
    boolean boxed,mesh,scalebox, displayxy,displayz,displaygrid;
    int calcdiv,dispdiv;
```

```
    if_xj f;
    public ExampleSurfaceModel2(if_xj f1,double xmini,double xmaxi,double ymini,double ymaxi,double zmini,double zmaxi)
                {f=f1;xmin=(float)xmini;xmax=(float)xmaxi;ymin=(float)ymini;ymax=(float)ymaxi;
                 zmin=(float)zmini;zmax=(float)zmaxi;
                 boxed=true;
                 mesh=true;
                 scalebox=false;
                 displayxy=true;
                 displayz=true;
                 displaygrid=true;
                 calcdiv=80;
                 dispdiv=80;
                 xlabel="X";
                 ylabel="Y";
                 zlabel="Z";
                }
    public ExampleSurfaceModel2(if_xj f1,double xmini,double xmaxi,double ymini,double ymaxi,double zmini,double zmaxi,String
lx,String ly,String lz)
                {f=f1;xmin=(float)xmini;xmax=(float)xmaxi;ymin=(float)ymini;ymax=(float)ymaxi;
                 zmin=(float)zmini;zmax=(float)zmaxi;
                 xlabel=lx;
                 ylabel=ly;
                 zlabel=lz;
                 boxed=true;
                 mesh=true;
                 scalebox=false;
                 displayxy=true;
                 displayz=true;
                 displaygrid=true;
                 calcdiv=100;
                 dispdiv=100;
                }
                public float calculateZ(float x, float y)
                {   double z[]={x,y};
                            return (float)f.func(z);
                }

    public double calculateZ(double x, double y)
    {           double z[]={x,y};
                   return f.func(z);}
    public double calculateZ(double z[])
    {           return f.func(z);
                }
                public int getPlotMode()
                {return ISurfacePlotModel.PLOT_MODE_SPECTRUM;}

                public boolean isBoxed()
                {return boxed;}

                public boolean isMesh()
                {return mesh;}

                public boolean isScaleBox()
                {return scalebox;}

                public boolean isDisplayXY()
                {return displayxy;}

                public boolean isDisplayZ()
                {return displayz;}

                public boolean isDisplayGrids()
                {return displaygrid;}

                public void setCalcDivisions(int calcdivi )
                {calcdiv=calcdivi;}
                public int getCalcDivisions()
                {return calcdiv;}
    public void setDispDivisions(int dispdivi )
                {dispdiv=dispdivi;}
                public int getDispDivisions()
                {return dispdiv;}
    public void setXMin(double xmini)
                {xmin=(float)xmini;}
                public float getXMin()
```

```
        {return xmin;}
    public void setXMax(double xmaxi)
        {xmax=(float)xmaxi;}
        public float getXMax()
        {return xmax;}
    public void setYMin(double ymini)
        {ymin=(float)ymini;}
        public float getYMin()
        {return ymin;}
    public void setYMax(double ymaxi)
        {ymax=(float)ymaxi;}
        public float getYMax()
        {return ymax;}
        public void setZMin(double zmini)
        {zmin=(float)zmini;}
        public float getZMin()
        {return zmin;}
        public void setZMax(double zmaxi)
        {zmax=(float)zmaxi;}
        public float getZMax()
        {return zmax;}
        public void setXAxisLabel(String xlabeli)
        {xlabel=xlabeli;}
        public String getXAxisLabel()
        {return xlabel;}
        public void setYAxisLabel(String ylabeli)
        {ylabel=ylabeli;}
        public String getYAxisLabel()
        {return ylabel;}
        public void setZAxisLabel(String zlabeli)
        {zlabel=zlabeli;}
        public String getZAxisLabel()
        {return zlabel;}
}
```

EX5:



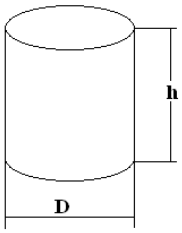
In order to make a cartoon box, the shape above should be used.After bending from the dotted line the volume of the box will be 1 m^3. In order to spent the minimum amount of the cartoon, what a and b dimensions should be?

EX 6: Find the minimum of function

$$f(x) = \frac{15x}{(4x^2 - 3x + 4)}$$ f is in the range of 0 to10.

HOMEWORK EXERCISES

Homework exercises will be done at home and will bring to next friday class printed no late exercises will be excepted.



W5HW1:

One of the very basic optimization problem is the minimum cost of container problem. The cost of a box usually is a function of the surface area. Therefore we should minimize the area for a given volume For example if the volume of the container

V=0.5 liter=0.5x10^{-3} m^3 :

Volume $V = \frac{\pi D^2}{4} h$ or from this equation h, height is obtained as $h = \frac{4V}{\pi D^2}$.

Surface area of the cylinder : $A = 2\frac{\pi D^2}{4} + \pi D h = \frac{\pi D^2}{2} + \frac{4V}{D}$.

Analytical solution of the minimization problem

$$\frac{dA}{dD} = \pi D - \frac{4V}{D^2} = 0$$

$D = \sqrt[3]{\frac{4V}{\pi}}$. From here solution is $D = \sqrt[3]{\frac{4*0.5x10^{-3}}{\pi}}$ =0.086025401 m and .

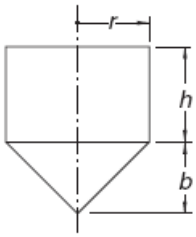
$h = \frac{4*0.5x10^{-3}}{\pi D^2}$ = 0.086025401 m.

Now obtain this results by using numerical optimization methods.

For the range of 0.01<=D<=0.2

a) Graphic method
b) Golden ratio (Fibonnachi)
c) Secant method
d) Second degree polynomials

W5HW2:



It is desired to have volume of the cone base shape shown in the figure as 1 m^3. Calculate r,h and b dimensions to minimise the surface area.

$$V = \pi r^2 \left(\frac{b}{3} + h \right)$$

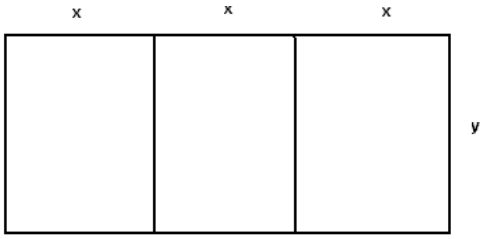
$$S = \pi r \left(2h + \sqrt{b^2 + r^2} \right)$$

W5HW3 :

Find the minimum of

$$f = 25x^2 - 12x^4 + 6xy + 25y^2 - 24x^2y^2 - 12y^4$$

W3HW4: A farmer needs 150 meters of fencing to fence three adjacent gardens. What would be maximum area of each garden, and what would be values of x and y



$4y + 6x = 150$

$y = 37.5 - 1.5x$

$A(x) = 3(37.5 - 1.5x)x = 112.5x - 4.5x^2$

$\frac{dA(x)}{dx} = 112.5 - 9x = 0$

$x = 12.5$ y=18.75

$A = 703.125$ total of 3 garden

Calculate by using a numerical method