```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import java.util.*;
import java.net.*;
import javax.imageio.*;
import java.awt.image.*;

public class Bem3DMesh extends JApplet implements ActionListener, ItemListener {

	protected String
chart,dir,file_chart,file_jpg,file_bound,file_depth,file_gif,file_tridata,file_squdata;
	protected String str,file_name ;
	protected String
file_left,file_right,file_middle,file_surface,file_seabed,file_quay;

	protected int kbd1,kcm1,lte1,kte1,ked1,nob1,nib1,inp,iq,totalN1,boundN1;
	protected int kbd,kcm,lte,kte,ked,nin,nob,nib,node;
	protected int ntp,ia,ib,ic,i,j,k,nb,np,ma,mb,mc,ms,mp,loc,it,itop,maxstk,ip,itri;
	protected int iv1,iv2,iv3,iedge,il,ir,iera,ierb,ierl,iswap;
	protected int ja,jb,mstk,nbr,idf,ielm,jelm,kelm,min,l,jr,npa,npb,nta,ntb;
	protected int ivx,ips,ipg,iva,ivb,jvx,nnpl,nbs,ix,ncount;
	protected int jl1,jl2,jl3,jr1,jr2,jr3,iv4,jz,js,ndiv;
	protected int iz,inelm,lelm,ikp,kedg,ledg,inb,ka,kb,la,lb,itera,j1,j2,j3,j4;
	protected int iv,jn,ie1,ie2,ie3,jacia,jacib,inn1,inn2,ien1,ien2;
	protected int nodt,nelt,ne,kpt,kxx1, kyy1,kxx2,kyy2,kxx3,kyy3,kxx4,kyy4;
	protected int hsize,vsize,nod1,nod2,nod3,nod4,check;
	protected int m,n,ien,kxi,kxf,kyi,kyf,axis,xtemp,ytemp,eps_num0;
	protected int xx,yy,xc,yc,ii,ij,jj,x_plot,y_plot,lengthh;
	protected int xpr1,ypr1,xpr2,ypr2,xpr3,ypr3,xpr4,ypr4,ktj,ni_depth,nj_depth;

	protected int xgenten,ygenten,crtgx,crtgy,xcmin,xcmax,ycmin,ycmax;
	protected int xx1,yy1,xx2,yy2,totalInn,ibex,iii;
	protected int xl,yl,xr,yr;
	protected int igrid =0;
	protected int work    =0;
```

```
        protected int nelm1,nelm2,nelm3,nelm4,nelm5,nelm6,nelmm,jg;
        protected int node1,node2,node3,node4,node5,node6;
        protected int dki      =1;
        protected int procedure=0;
        protected int dot=3;
        protected int nex=6;
        protected double alpha=3.0;
        protected double pi=Math.PI;
        protected double pi2=pi*2;

        protected int rightcoastStart;
        protected int rightcoastEnd;
        protected int leftcoastStart;
        protected int leftcoastEnd;
        protected int quayStart;
        protected int quayEnd;
        protected int rightcoastNum=0;
        protected int leftcoastNum=0;
        protected int islandNumm=0;
        protected int quayNum=0;
        protected int innerNode=0;
        protected int innerPoint=0;
        protected int innNodeStart=0;
        protected int innNodeEnd=0;
        protected int innPointStart=0;
        protected int innPointEnd=0;
        protected int innPointNum=0;
        protected int innNodeNum=0;
        protected int
tempN,pseudoN1,pseudoN2,pseudoN3,pseudoN,clockwise,gridinterval;

        protected double r,r1,r2,griddata,dx0,dy0,z_xy,ds,sqarea;
        protected double long_chart,lati_chart,long_diag,lati_diag;
        protected double x_length,y_length,mscale,dep,depth_xy;
        protected double x1,x2,y1,y2,xkgein,ykgein;
        protected double dx,dy,yi,xj,xtmp,ytmp,s,xxl,yyl,xy_z,quayH;
```

```
    protected    PrintWriter pw_bound;
    protected    PrintWriter pw_chart;
    protected    PrintWriter pw_depth;
    protected    PrintWriter pw_tridata;
    protected    PrintWriter pw_squdata;

    protected    BufferedReader br_bound;
    protected    BufferedReader br_chart;
    protected    BufferedReader br_depth;
    protected    BufferedReader br_tridata;
    protected    BufferedReader br_squdata;

    protected double
xmin,xmax,ymin,ymax,rax,ray,dmax,pxx,pyy,x13,y13,x23,y23,x1p,y1p,x2p,y2p;
    protected double cosa,cosb,sina,sinb,gx,gy,ar;
    protected double cgrax,cgray,a1,a2,range,pa1,pb1,pa2,pb2,pa3,pb3,pa4,pb4;
    protected double
dltd,delta,xg,yg,xs,ys,xp,yp,xa,ya,xb,yb,see,pxa,pya,pxb,pyb,pxc,pyc;
    protected double px12,py12,px23,py23,pl12,pl23;
    protected double dpp,eps,eps0,epl;
    protected double xx4,yy4,xc4,yc4,depthIndex ;
    protected double amean,amin,amax;

    protected int []islandNum;

    protected int [][]ihen;
    protected int [][]kakom;
    protected int [][]iena;
    protected int [][]ienb;
    protected int [][]jeea;
    protected int [][]jeeb;
    protected int [][]jac1;
    protected int [][]jac2;
    protected int [][]jac3;
    protected int [][]jac4;
    protected int [][]jac5;
    protected int [][]jac6;
    protected int []ibno1;
```

```
protected int [][]mtj1;
protected int [][]mtj2;
protected int [][]mtj3;
protected int [][]mtj4;
protected int [][]mtj5;
protected int [][]mtj6;
protected int [][]nei1;
protected int [][]nei2;
protected int [][]nei3;
protected int [][]nei4;
protected int [][]nei5;
protected int [][]nei6;
protected int []jnb1;
protected int []jnb2;
protected int []jnb3;
protected int []jnb4;
protected int []jnb5;
protected int []jnb6;
protected double []xn1;
protected double []xn2;
protected double []xn3;
protected double []xn4;
protected double []xn5;
protected double []xn6;
protected double []yn1;
protected double []yn2;
protected double []yn3;
protected double []yn4;
protected double []yn5;
protected double []yn6;
protected double []zn1;
protected double []zn2;
protected double []zn3;
protected double []zn4;
protected double []zn5;
protected double []zn6;
protected double []px1;
```

```
protected double []py1;
protected double []pz1;
protected double []px2;
protected double []py2;
protected double []pz2;
protected double []px3;
protected double []py3;
protected double []pz3;
protected double []px4;
protected double []py4;
protected double []pz4;
protected double []px5;
protected double []py5;
protected double []pz5;
protected double []px6;
protected double []py6;
protected double []pz6;
protected double []area1;
protected double []area2;
protected double []area3;
protected double []area4;
protected double []area5;
protected double []area6;

protected int [][]mmtj;
protected int []id;
protected int []nsra;
protected int []nsrb;
protected int []jhen;
protected int []iad;
protected int []jstack;
protected int []matno;
protected int []khen;
protected int []ibin;
protected int []index;
protected int []idm;
protected int []nedg;
protected int []kstack;
```

```
        protected int []list;
        protected int []istack;
        protected int []kv;
        protected int []ibr;
        protected int []iadres;
        protected int []map;
        protected int []ifix;
        protected int []islandStart;
        protected int []islandEnd;
        protected double [][]gridDepth;
        protected double []angl;
        protected double delx;
        protected double []datax;
        protected double []datay;
        protected int []mkp=new int[3];
        protected int []jkp=new int[3];
        protected double []pl=new double[3];

        Container gcp;
        Graphics g;
        Image img;
        Figure figure;

        JLabel
lb_chart,lb_dir,lb_enter,lb_nodt,lb_surfaceN,lb_clockwise,lb_depthIndex,lb_quayH;
        JLabel
lb_command,lb_command1,lb_node,lb_x,lb_y,lb_z,lb_kpt,lb_lengthh;
        JLabel
lb_nin,lb_nob,lb_nib,lb_ibex,lb_kbd,lb_ktj,lb_kcm,lb_lte,lb_gridinterval;

        JTextField
tf_title,tf_chart,tf_dir,tf_nodt,tf_surfaceN,tf_clockwise,tf_depthIndex,tf_quayH;
        JTextField tf_command,tf_command1,tf_node,tf_x,tf_y,tf_z,tf_kpt,tf_lengthh;
        JTextField tf_nin,tf_nob,tf_nib,tf_ibex,tf_kbd,tf_ktj,tf_kcm,tf_lte,tf_gridinterval;

        JButton basin,island,innnode,innpoint,end,all_end,pseudo,squreloc,waterdepth;
        JButton enter,trimesh,triplot,squmesh,squplot,save,quayh,rightcoast,leftcoast;
```

```
    Choice ch_chart,ch_help;

public void init(){

        enter=null;innpoint=null;
        basin=null;island=null;end=null;all_end=null;

        gcp=getContentPane();
        gcp.setLayout(null);
        gcp.setForeground(Color.black);
        gcp.setBackground(Color.white);

        ch_help = new Choice();
        ch_help.setBounds(20,0,850,20);
        ch_help.setForeground(Color.black);
        ch_help.setBackground(Color.white);
        ch_help.setVisible(true);
        ch_help.add("
使          用          說          明    ");
        ch_help.add(" 1. 應用海域水深及港形生成取得水深及港形數據");
        ch_help.add(" 2. 建立存放檔案子目錄(例如 D:/chart),複製圖像至此目
錄");
        ch_help.add(" 3. 輸入海圖影像檔(英文)主檔名(但副檔名為 jpg)海圖影
像解析度可隨使用者顯示而異,可利用繪圖軟體適度調整");
        ch_help.add(" 4. 3 項為離島數,4-8 項為概估值,目的為宣告陣列大小,若
設定值小於將來實際值時會出現錯誤,重設即可。");
        ch_help.add(" 5. 放大倍率可取任意整數,陸地判定值的目的是自動削除
陸地上網格節點,把極鄰近邊界節點一併刪除,取負值為宜");
        ch_help.add(" 6. 在海圖(紙本)繪出港口外海假想邊界範圍並決定 3 個
假想邊界位置(參考 pdf 說明)");
        ch_help.add(" 6. 輸入第 1,第 2,第 3 假想邊界線節點數,視實際需要決定
後按確認");
        ch_help.add(" ");
        ch_help.add(" 7. 按「設定假想邊界端點」,輸入方向一般 FEM 採順時
針,BEM 採逆時針");
        ch_help.add("      順時針時第 1 假想邊界端點在左下角,第 2 點在左上,
第 3 點在右上,第 4 點在右下角,依序按滑鼠設定。");
        ch_help.add("      逆時針時第 1 假想邊界端點在右下角,第 2 點在右上,
```

第 3 點在左上,第 4 點在左下角,依序按滑鼠設定。");
ch_help.add(" 設定 4 個假想邊界線節點後按「個案結束」(不理想時請重設)。");
ch_help.add(" 8. 按「外部邊界」設定港形邊界節點,從第 4 假想邊界端點近旁開始,");
ch_help.add(" 以順(逆)時針方向依序按滑鼠設定,於設定最後 l 個節點前(即與第 4 假想邊界端點相鄰時)");
ch_help.add(" 先按「個案結束」後,再按滑鼠決定最後節點,目的為使邊界閉合。");
ch_help.add(" 9. 若有離島,按「離島」,可從任意點開始按滑鼠設定離島邊界上的節點,但其方向必須與外部邊界的方向相反,");
ch_help.add(" 當設定至鄰近起始點的鄰近最後 1 個時,先按「個案選點結束」後,再按滑鼠決定最後節點");
ch_help.add("10. 若有多個離島重複步驟 9 ");
ch_help.add("11. 若需要設置內部網格節點時按「內部網格節點」,其目的為元素排列整齊化");
ch_help.add("12. 若有特殊需要加強時可於領域內任何位置設置節點,此時按「內部加強節點」設置節點,重複完成後按「個案選點結束」");

ch_help.add("13. 按「全部選點結束」 ");
ch_help.add("14. 按「四邊形元素生成」");
ch_help.add("15. 若欲修正元素形狀時按「四邊形元素修正」");
ch_help.add("16. 按「四邊形繪圖」");
ch_help.add(" ");
//ch_help.add(" 完成的圖檔及數據分別存入 gif 及 data 檔 ");
ch_help.addItemListener(this);
gcp.add(ch_help);

lb_dir = new JLabel("1.海圖子目錄名");
lb_dir.setBounds(0,40,100,20);
lb_dir.setForeground(Color.black);
lb_dir.setBackground(Color.white);
lb_dir.setVisible(true);
gcp.add(lb_dir);

tf_dir = new JTextField("");
tf_dir.setBounds(100,40,120,20);
tf_dir.setForeground(Color.black);

```
tf_dir.setBackground(Color.white);
tf_dir.setEditable(true);
tf_dir.setVisible(true);
gcp.add(tf_dir);

lb_chart = new JLabel("2.海圖名(英文)");
lb_chart.setBounds(0,60,100,20);
lb_chart.setForeground(Color.black);
lb_chart.setBackground(Color.white);
lb_chart.setVisible(true);
gcp.add(lb_chart);

tf_chart = new JTextField("");
tf_chart.setBounds(100,60,120,20);
tf_chart.setForeground(Color.black);
tf_chart.setBackground(Color.white);
tf_chart.setEditable(true);
tf_chart.setVisible(true);
gcp.add(tf_chart);

lb_nin = new JLabel("3.離島數");
lb_nin.setBounds(0,80,100,20);
lb_nin.setForeground(Color.black);
lb_nin.setBackground(Color.white);
gcp.add(lb_nin);

tf_nin = new JTextField("0");
tf_nin.setBounds(100,80,120,20);
tf_nin.setForeground(Color.black);
tf_nin.setBackground(Color.white);
tf_nin.setEditable(true);
gcp.add(tf_nin);

lb_nodt = new JLabel("4.概估總節點數");
lb_nodt.setBounds(0,100,100,20);
lb_nodt.setForeground(Color.black);
lb_nodt.setBackground(Color.white);
lb_nodt.setVisible(true);
```

```
gcp.add(lb_nodt);

tf_nodt = new JTextField("20000");
tf_nodt.setBounds(100,100,120,20);
tf_nodt.setForeground(Color.black);
tf_nodt.setBackground(Color.white);
tf_nodt.setEditable(true);
tf_nodt.setVisible(true);
gcp.add(tf_nodt);

lb_kbd = new JLabel("5.概估境界節點數");
lb_kbd.setBounds(250,40,160,20);
lb_kbd.setForeground(Color.black);
lb_kbd.setBackground(Color.white);
lb_kbd.setVisible(true);
gcp.add(lb_kbd);

tf_kbd = new JTextField("5000");
tf_kbd.setBounds(410,40,60,20);
tf_kbd.setForeground(Color.black);
tf_kbd.setBackground(Color.white);
tf_kbd.setEditable(true);
tf_kbd.setVisible(true);
gcp.add(tf_kbd);

lb_ktj = new JLabel("6.概估最多節點數");
lb_ktj.setBounds(250,60,160,20);
lb_ktj.setForeground(Color.black);
lb_ktj.setBackground(Color.white);
lb_ktj.setVisible(true);
gcp.add(lb_ktj);

tf_ktj = new JTextField("30000");
tf_ktj.setBounds(410,60,60,20);
tf_ktj.setForeground(Color.black);
tf_ktj.setBackground(Color.white);
tf_ktj.setEditable(true);
tf_ktj.setVisible(true);
```

```
gcp.add(tf_ktj);

lb_kcm = new JLabel("7.概估節點集合最多元素數");
lb_kcm.setBounds(250,80,160,20);
lb_kcm.setForeground(Color.black);
lb_kcm.setBackground(Color.white);
lb_kcm.setVisible(true);
gcp.add(lb_kcm);

tf_kcm = new JTextField("8");
tf_kcm.setBounds(410,80,60,20);
tf_kcm.setForeground(Color.black);
tf_kcm.setBackground(Color.white);
tf_kcm.setEditable(true);
tf_kcm.setVisible(true);
gcp.add(tf_kcm);

lb_lte = new JLabel("8.概估多角形的最大邊數");
lb_lte.setBounds(250,100,160,20);
lb_lte.setForeground(Color.black);
lb_lte.setBackground(Color.white);
lb_lte.setVisible(true);
gcp.add(lb_lte);

tf_lte = new JTextField("10");
tf_lte.setBounds(410,100,60,20);
tf_lte.setForeground(Color.black);
tf_lte.setBackground(Color.white);
tf_lte.setEditable(true);
tf_lte.setVisible(true);
gcp.add(tf_lte);

lb_clockwise = new JLabel("9.順時針=1,逆時針=2");
lb_clockwise.setBounds(480,40,140,20);
lb_clockwise.setForeground(Color.black);
lb_clockwise.setBackground(Color.white);
lb_clockwise.setVisible(true);
gcp.add(lb_clockwise);
```

```
tf_clockwise = new JTextField("1");
tf_clockwise.setBounds(620,40,60,20);
tf_clockwise.setForeground(Color.black);
tf_clockwise.setBackground(Color.white);
tf_clockwise.setEditable(true);
tf_clockwise.setVisible(true);
gcp.add(tf_clockwise);

lb_kpt = new JLabel("10.放大倍率");
lb_kpt.setBounds(480,60,140,20);
lb_kpt.setForeground(Color.black);
lb_kpt.setBackground(Color.white);
lb_kpt.setVisible(true);
gcp.add(lb_kpt);

tf_kpt = new JTextField("1");
tf_kpt.setBounds(620,60,60,20);
tf_kpt.setForeground(Color.black);
tf_kpt.setBackground(Color.white);
tf_kpt.setEditable(true);
tf_kpt.setVisible(true);
gcp.add(tf_kpt);

lb_depthIndex = new JLabel("11.陸地判定値");
lb_depthIndex.setBounds(480,80,140,20);
lb_depthIndex.setForeground(Color.black);
lb_depthIndex.setBackground(Color.white);
lb_depthIndex.setVisible(true);
gcp.add(lb_depthIndex);

tf_depthIndex = new JTextField("-2");
tf_depthIndex.setBounds(620,80,60,20);
tf_depthIndex.setForeground(Color.black);
tf_depthIndex.setBackground(Color.white);
tf_depthIndex.setEditable(true);
tf_depthIndex.setVisible(true);
gcp.add(tf_depthIndex);
```

```
lb_lengthh = new JLabel("12.繪圖寬");
lb_lengthh.setBounds(480,100,140,20);
lb_lengthh.setForeground(Color.black);
lb_lengthh.setBackground(Color.white);
lb_lengthh.setVisible(true);
gcp.add(lb_lengthh);

tf_lengthh = new JTextField("500");
tf_lengthh.setBounds(620,100,60,20);
tf_lengthh.setForeground(Color.black);
tf_lengthh.setBackground(Color.white);
tf_lengthh.setEditable(true);
tf_lengthh.setVisible(true);
gcp.add(tf_lengthh);

lb_gridinterval = new JLabel("13.內部網格間隔");
lb_gridinterval.setBounds(690,40,110,20);
lb_gridinterval.setForeground(Color.black);
lb_gridinterval.setBackground(Color.white);
lb_gridinterval.setVisible(true);
gcp.add(lb_gridinterval);

tf_gridinterval = new JTextField("1");
tf_gridinterval.setBounds(800,40,40,20);
tf_gridinterval.setForeground(Color.black);
tf_gridinterval.setBackground(Color.white);
tf_gridinterval.setEditable(true);
tf_gridinterval.setVisible(true);
gcp.add(tf_gridinterval);

lb_enter = new JLabel("輸入 1-13 項後,按確認");
lb_enter.setBounds(720,65,120,20);
lb_enter.setForeground(Color.red);
lb_enter.setBackground(Color.white);
gcp.add(lb_enter);

enter = new JButton("確                    認");
```

```
enter.setBounds(720,85,120,25);
enter.setForeground(Color.black);
enter.setBackground(Color.cyan);
enter.addActionListener(this);
gcp.add(enter);

lb_command = new JLabel("執行事項");
lb_command.setBounds(0,120,60,20);
lb_command.setForeground(Color.blue);
lb_command.setBackground(Color.white);
gcp.add(lb_command);

tf_command = new JTextField(" ");
tf_command.setBounds(60,120,120,20);
tf_command.setForeground(Color.black);
tf_command.setBackground(Color.white);
tf_command.setEditable(true);
gcp.add(tf_command);

lb_command1 = new JLabel("執行工作");
lb_command1.setBounds(190,120,60,20);
lb_command1.setForeground(Color.blue);
lb_command1.setBackground(Color.white);
gcp.add(lb_command1);

tf_command1 = new JTextField("          ");
tf_command1.setBounds(250,120,400,20);
tf_command1.setForeground(Color.red);
tf_command1.setBackground(Color.white);
tf_command1.setVisible(true);
tf_command1.setEditable(true);
gcp.add(tf_command1);

lb_node = new JLabel("節點編號");
lb_node.setBounds(0,150,60,20);
lb_node.setForeground(Color.green);
lb_node.setBackground(Color.white);
lb_node.setVisible(true);
```

```
gcp.add(lb_node);

tf_node = new JTextField("");
tf_node.setBounds(60,150,60,20);
tf_node.setForeground(Color.black);
tf_node.setBackground(Color.white);
tf_node.setEditable(true);
tf_node.setVisible(true);
gcp.add(tf_node);

lb_x = new JLabel("x 座標");
lb_x.setBounds(120,150,40,20);
lb_x.setForeground(Color.green);
lb_x.setBackground(Color.white);
gcp.add(lb_x);

tf_x = new JTextField("");
tf_x.setBounds(160,150,100,20);
tf_x.setForeground(Color.black);
tf_x.setBackground(Color.white);
tf_x.setEditable(true);
gcp.add(tf_x);

lb_y = new JLabel("y 座標");
lb_y.setBounds(260,150,40,20);
lb_y.setForeground(Color.green);
lb_y.setBackground(Color.white);
gcp.add(lb_y);

tf_y = new JTextField("");
tf_y.setBounds(300,150,100,20);
tf_y.setForeground(Color.black);
tf_y.setBackground(Color.white);
tf_y.setEditable(true);
gcp.add(tf_y);

lb_z = new JLabel("水深");
lb_z.setBounds(400,150,30,20);
```

```
lb_z.setForeground(Color.green);
lb_z.setBackground(Color.white);
gcp.add(lb_z);

tf_z = new JTextField("");
tf_z.setBounds(430,150,50,20);
tf_z.setForeground(Color.black);
tf_z.setBackground(Color.white);
tf_z.setEditable(true);
gcp.add(tf_z);

lb_quayH = new JLabel("岸壁出水高度");
lb_quayH.setBounds(500,150,80,20);
lb_quayH.setForeground(Color.black);
lb_quayH.setBackground(Color.white);
gcp.add(lb_quayH);

tf_quayH = new JTextField("0");
tf_quayH.setBounds(580,150,40,20);
tf_quayH.setForeground(Color.black);
tf_quayH.setBackground(Color.white);
tf_quayH.setEditable(true);
gcp.add(tf_quayH);

quayh = new JButton("設定岸壁高");
quayh.setBounds(620,150,100,20);
quayh.setForeground(Color.black);
quayh.setBackground(Color.cyan);
quayh.addActionListener(this);
quayh.setVisible(true);
gcp.add(quayh);

lb_ibex = new JLabel("外部邊界節點數");
lb_ibex.setBounds(0,190,100,20);
lb_ibex.setForeground(Color.red);
lb_ibex.setBackground(Color.white);
lb_ibex.setVisible(true);
gcp.add(lb_ibex);
```

```
tf_ibex = new JTextField("");
tf_ibex.setBounds(100,190,120,20);
tf_ibex.setForeground(Color.black);
tf_ibex.setBackground(Color.white);
tf_ibex.setEditable(true);
tf_ibex.setVisible(true);
gcp.add(tf_ibex);

lb_nob = new JLabel("全部邊界節點數");
lb_nob.setBounds(220,190,100,20);
lb_nob.setForeground(Color.red);
lb_nob.setBackground(Color.white);
lb_nob.setVisible(true);
gcp.add(lb_nob);

tf_nob = new JTextField("");
tf_nob.setBounds(320,190,60,20);
tf_nob.setForeground(Color.black);
tf_nob.setBackground(Color.white);
tf_nob.setEditable(true);
tf_nob.setVisible(true);
gcp.add(tf_nob);

lb_nib = new JLabel("內部節點數");
lb_nib.setBounds(380,190,80,20);
lb_nib.setForeground(Color.red);
lb_nib.setBackground(Color.white);
lb_nib.setVisible(true);
gcp.add(lb_nib);

tf_nib = new JTextField("");
tf_nib.setBounds(460,190,60,20);
tf_nib.setForeground(Color.black);
tf_nib.setBackground(Color.white);
tf_nib.setEditable(true);
tf_nib.setVisible(true);
gcp.add(tf_nib);
```

```
end = new JButton("個案選點結束");
end.setBounds(550,190,120,20);
end.setForeground(Color.black);
end.setBackground(Color.cyan);
end.addActionListener(this);
end.setVisible(true);
gcp.add(end);

all_end = new JButton("全部選點結束");
all_end.setBounds(670,190,120,20);
all_end.setForeground(Color.black);
all_end.setBackground(Color.cyan);
all_end.addActionListener(this);
all_end.setVisible(true);
gcp.add(all_end);

pseudo = new JButton("1  設定假想邊界線端點");
pseudo.setBounds(0,220,170,20);
pseudo.setForeground(Color.black);
pseudo.setBackground(Color.cyan);
pseudo.addActionListener(this);
pseudo.setVisible(true);
gcp.add(pseudo);

rightcoast = new JButton("2  右側海岸線");
rightcoast.setBounds(170,220,120,20);
rightcoast.setForeground(Color.black);
rightcoast.setBackground(Color.cyan);
rightcoast.addActionListener(this);
rightcoast.setVisible(true);
gcp.add(rightcoast);

basin = new JButton("3  港形");
basin.setBounds(290,220,90,20);
basin.setForeground(Color.black);
basin.setBackground(Color.cyan);
basin.addActionListener(this);
```

```
basin.setVisible(true);
gcp.add(basin);

leftcoast = new JButton("4  左側海岸線");
leftcoast.setBounds(380,220,120,20);
leftcoast.setForeground(Color.black);
leftcoast.setBackground(Color.cyan);
leftcoast.addActionListener(this);
leftcoast.setVisible(true);
gcp.add(leftcoast);

island = new JButton("5  離島");
island.setBounds(500,220,90,20);
island.setForeground(Color.black);
island.setBackground(Color.cyan);
island.addActionListener(this);
island.setVisible(true);
gcp.add(island);

innnode = new JButton("6  內部網格點");
innnode.setBounds(590,220,120,20);
innnode.setForeground(Color.black);
innnode.setBackground(Color.cyan);
innnode.addActionListener(this);
innnode.setVisible(true);
gcp.add(innnode);

innpoint = new JButton("7  內部加強點");
innpoint.setBounds(710,220,120,20);
innpoint.setForeground(Color.black);
innpoint.setBackground(Color.cyan);
innpoint.addActionListener(this);
innpoint.setVisible(true);
gcp.add(innpoint);

trimesh = new JButton("8  三角形元素生成");
trimesh.setBounds(0,250,140,20);
trimesh.setForeground(Color.black);
```

```
trimesh.setBackground(Color.cyan);
trimesh.addActionListener(this);
trimesh.setVisible(true);
//gcp.add(trimesh);

triplot = new JButton("9  三角形繪圖");
triplot.setBounds(140,250,120,20);
triplot.setForeground(Color.black);
triplot.setBackground(Color.cyan);
triplot.addActionListener(this);
triplot.setVisible(true);
//gcp.add(triplot);

squmesh = new JButton("8  四邊形元素生成");
squmesh.setBounds(260,250,140,20);
squmesh.setForeground(Color.black);
squmesh.setBackground(Color.cyan);
squmesh.addActionListener(this);
squmesh.setVisible(true);
gcp.add(squmesh);

waterdepth = new JButton("9  水深");
waterdepth.setBounds(400,250,60,20);
waterdepth.setForeground(Color.black);
waterdepth.setBackground(Color.cyan);
waterdepth.addActionListener(this);
waterdepth.setVisible(true);
//gcp.add(waterdepth);

squreloc = new JButton("9  四邊形元素修正");
squreloc.setBounds(400,250,150,20);
squreloc.setForeground(Color.black);
squreloc.setBackground(Color.cyan);
squreloc.addActionListener(this);
squreloc.setVisible(true);
gcp.add(squreloc);

squplot = new JButton(" 10  四邊形繪圖");
```

```
            squplot.setBounds(550,250,120,20);
            squplot.setForeground(Color.black);
            squplot.setBackground(Color.cyan);
            squplot.addActionListener(this);
            squplot.setVisible(true);
            gcp.add(squplot);

            save = new JButton("11  存檔");
            save.setBounds(730,250,90,20);
            save.setForeground(Color.black);
            save.setBackground(Color.cyan);
            save.addActionListener(this);
            save.setVisible(true);
            gcp.add(save);

} //init
      public void itemStateChanged(ItemEvent e) { }
      public void actionPerformed(ActionEvent ae) {
         String str;

         if(enter==ae.getSource()){
                        pseudo.setVisible(true);
                rightcoast.setVisible(false);
                         basin.setVisible(false);
                  leftcoast.setVisible(false);
                       island.setVisible(false);
                    innnode.setVisible(false);
                   innpoint.setVisible(false);
                            end.setVisible(false);
                    all_end.setVisible(false);
                    trimesh.setVisible(false);
                    triplot.setVisible(false);
                    squmesh.setVisible(false);
                  squreloc.setVisible(false);
                    squplot.setVisible(false);

            ReadChart();
```

```
        //nodt=Integer.valueOf(tf_nodt.getText().trim()).intValue();
        nin =Integer.valueOf(tf_nin.getText().trim()).intValue();
        kbd =Integer.valueOf(tf_kbd.getText().trim()).intValue();
        ktj =Integer.valueOf(tf_ktj.getText().trim()).intValue();
        kcm =Integer.valueOf(tf_kcm.getText().trim()).intValue();
        lte =Integer.valueOf(tf_lte.getText().trim()).intValue();
        nodt=Integer.valueOf(tf_nodt.getText().trim()).intValue();
        kpt =Integer.valueOf(tf_kpt.getText().trim()).intValue();
        clockwise=Integer.valueOf(tf_clockwise.getText().trim()).intValue();
        depthIndex =Double.valueOf(tf_kpt.getText().trim()).doubleValue();
        lengthh =Integer.valueOf(tf_lengthh.getText().trim()).intValue();
        gridinterval =Integer.valueOf(tf_gridinterval.getText().trim()).intValue();
        quayH = Double.valueOf(tf_quayH.getText().trim()).doubleValue();

        ked=2*ktj+1;
        kte=2*ktj+1;

gridDepth=new double[ni_depth][nj_depth];
        ibno1=new int[ktj];                    //外部境界上的節點編號
          mtj1=new int[2*ktj+1][9];          //要素-節關係
          jac1=new int[2*ktj+1][4];          //要素-鄰接要素關係
        mmtj=new int[2*ktj+1][4];
        ihen=new int[2*lte+1][2];
           id=new int[2*ktj+1];
        nei1=new int[ktj][kcm];
          xn2=new double[ktj];
          yn2=new double[ktj];
          zn2=new double[ktj];

 for(i=1;i<=ktj;i++){for(j=1;j<=kcm;j++){nei1[i][j]=0;}}

        ifix=new int[ktj];
       index=new int[kbd+1];                  //作業領域
        ibin=new int[kbd];
          idm=new int[2*ktj+1];              //要素-領域關係
        iena=new int[lte][3];
        ienb=new int[lte][3];
        jeea=new int[lte][3];
```

```
        jeeb=new int[lte][3];
        nsra=new int[lte+2];
        nsrb=new int[lte+2];
        jhen=new int[2*lte+1];
         iad=new int[2*lte+1];
     jstack=new int[lte+2];
        jnb1=new int[ktj];
        ifix=new int[ktj];
        nedg=new int[2*ktj+1];
     kstack=new int[2*ktj+1];
        list=new int[ktj];
     istack=new int[ktj];
          kv=new int[2*ktj+1];
        ibr=new int[2*ktj+1];
     iadres=new int[ktj+3];
         map=new int[2*ktj+1];

     islandEnd=new int[nin] ;
islandStart=new int[nin] ;
     islandNum=new int[nin] ;

      area1=new double[ktj+3];
        px1=new double[ktj+3];          //節點 x 座標
        py1=new double[ktj+3];          //節點 y 座標
        pz1=new double[ktj+3];
       angl=new double[2*ktj+1];

        ReadgridDepth();

        if(axis==1)    {
              x_length=Math.abs((long_diag-long_chart)*111000);
              y_length=Math.abs((lati_diag-lati_chart)*111000);}
        if(axis==2)    {
             x_length=Math.abs(long_diag-long_chart);
             y_length=Math.abs(lati_diag-lati_chart);}

        xkgein =x_length/crtgx;
        ykgein =y_length/crtgy;
```

```
            figure = new Figure();
            figure.setBounds(0,220,crtgx,crtgy);
            figure.setForeground(Color.black);
            figure.setBackground(Color.gray);
            gcp.add(figure);

            tf_command1.setText("");
        }   //enter

       if(pseudo==ae.getSource()){
             procedure = 1;
             check=0;
             JOptionPane.showMessageDialog(this, "按滑鼠設定假想邊界第 1 端點
");
                        pseudo.setVisible(false);
                  rightcoast.setVisible(true);
                           end.setVisible(true);}

        if(rightcoast==ae.getSource()){
               procedure = 2;
               rightcoastStart=pseudoN+1;
               rightcoast.setVisible(false);
               basin.setVisible(true); }

        if(basin==ae.getSource()){
               procedure = 3;
               quayStart=dki;
               basin.setVisible(false);
               leftcoast.setVisible(true);}

        if(leftcoast==ae.getSource()){
               procedure = 4;
               leftcoastStart=dki;
                  leftcoast.setVisible(false);
                       island.setVisible(true);
                    all_end.setVisible(true);}
```

```
        if(island==ae.getSource()){
                procedure = 5;
                islandNumm=islandNumm+1;
                islandStart[islandNumm]=dki;
                   leftcoast.setVisible(false);
                       island.setVisible(true);
                     all_end.setVisible(true);
                     innnode.setVisible(true);
                    innpoint.setVisible(true);
                            end.setVisible(true);
                     all_end.setVisible(true);}

        if(innnode==ae.getSource()){
                procedure = 6;
                check=0;
                innNodeStart=dki;
                JOptionPane.showMessageDialog(this, "在任意位置按滑鼠以便設定
內部網格節點");
                       innnode.setVisible(false);
                    innpoint.setVisible(true);
                     trimesh.setVisible(false);
                     triplot.setVisible(false);
                     squmesh.setVisible(false);
                    squreloc.setVisible(false);
                     squplot.setVisible(false);}

        if(innpoint==ae.getSource()){
                procedure = 7;
                check=0;
                innPointStart=dki;
                    innpoint.setVisible(false);}

       if(end==ae.getSource()){work=10;}

       if(all_end==ae.getSource()){
              work = 4;
              All_End();
```

```
                    end.setVisible(false);
                all_end.setVisible(false);
                trimesh.setVisible(false);
                triplot.setVisible(false);
                squmesh.setVisible(true);
               squreloc.setVisible(true);
                squplot.setVisible(true);}

        if(trimesh==ae.getSource()){
            //triangleMesh();
                trimesh.setVisible(false);
                triplot.setVisible(true);
                squmesh.setVisible(true);
                squplot.setVisible(true);}


        if(triplot==ae.getSource()){
           procedure =8;
              JOptionPane.showMessageDialog(this, "在任意位置按滑鼠以便繪元
素分佈圖");

                triplot.setVisible(false);
                squmesh.setVisible(true);
                 squreloc.setVisible(false);
                squplot.setVisible(false);}

        if(squmesh==ae.getSource()){
            squareMesh();
                squmesh.setVisible(false);
                 squreloc.setVisible(false);
                squplot.setVisible(true);}

        if(squreloc==ae.getSource()){
            SquareRelocation();
                squmesh.setVisible(false);
              squreloc.setVisible(false);
                squplot.setVisible(true);}
        if(squplot==ae.getSource()){
```

```
            procedure =9;
                JOptionPane.showMessageDialog(this, "在任意位置按滑鼠以便繪元
素分佈圖");
                    squplot.setVisible(false);}
          if(save==ae.getSource()){
             SquareWriteData();
          }
          if(quayh==ae.getSource()){
                        quayH =
Double.valueOf(tf_quayH.getText().trim()).doubleValue();
          }
    }    //actionPerformed

class Figure extends Canvas{

       Image doublebuff;
       Graphics offg;


  Figure(){

          xgenten = 0;                                                    //CRT x  原點
          ygenten = 0;                                                    //CRT y  原點
          xcmin = xgenten;
          xcmax = xcmin + crtgx - 1;
          ycmin = ygenten;
          ycmax = ygenten + crtgy - 1;

      }//Figure

public boolean mouseDown(Event ev,int xx,int yy) {

             xc=xx;    yc=yy;
             if(xc<0)          xc=0;
             if(xc>crtgx) xc=crtgx;
             if(yc<0)          yc=0;
             if(yc>crtgy) yc=crtgy;
```

```
            tf_x.setText(DoubleFormat.toString(xc,5)) ;
            tf_y.setText(DoubleFormat.toString(yc,5)) ;
            AnyDepth(xc,yc,z_xy);
            tf_z.setText(DoubleFormat.toString(z_xy,2)) ;

            repaint();

          return true;}

public void init(){doublebuff=createImage(crtgx,crtgy);
offg=doublebuff.getGraphics();}
public void paint(Graphics g){if(doublebuff == null) this.init();update(g);}
public void update(Graphics g){

      Graphics2D g2 = (Graphics2D)g;
      BufferedImage img = null;

      try{img = ImageIO.read(new File(file_jpg)); }
      catch(Exception e) {tf_chart.setText("result");img = null;}
      if(img==null){img=new
BufferedImage(getWidth(),getHeight(),BufferedImage.TYPE_INT_BGR);}
      Graphics2D offg = img.createGraphics();

        if(procedure<7) gridDraw();

        if(procedure==1) PseudoBoundary();
        if(procedure==2) RightCoast();
        if(procedure==3) Basin();
        if(procedure==4) LeftCoast();
        if(procedure==5) Island();
        if(procedure==6) InnerNode();
        if(procedure==7) InnerPoint();
        if(procedure==8) TrianglePlot(node1,nelm1,px1,py1,mtj1);

        if(procedure==9) {
            for(i=1;i<=6;i++){
                if(i==1) {file_name=file_surface;
SquarePlot(node1,nelm1,px1,py1,mtj1); }
```

```
                    if(i==2) {file_name=file_left;
SquarePlot(node2,nelm2,px2,py2,mtj2); }
                    if(i==3) {file_name=file_middle;
SquarePlot(node3,nelm3,px3,py3,mtj3); }
                    if(i==4) {file_name=file_right;
SquarePlot(node4,nelm4,px4,py4,mtj4); }
                    if(i==5) {file_name=file_quay;
SquarePlot(node5,nelm5,px5,py5,mtj5); }
                    if(i==6) {file_name=file_seabed;
SquarePlot(node6,nelm6,px6,py6,mtj6); }
                 }

            g2.drawImage(img,0,0,this);

            try {boolean result=ImageIO.write(img,"gif",new File(file_name));}
            catch(Exception e) {tf_chart.setText("result");}
        }
  }    //update

public void gridDraw(){
      if(igrid==0) {
            dx0=(crtgx/(double)nj_depth);
            dy0=(crtgy/(double)ni_depth);
            delx=dx0;
            if(dy0>delx) delx=dy0;
            for(i=2;i<=nj_depth;i+=2){    //繪網格
                  int xg=(int)(i*dx0);
                  offg.setColor(Color.black);
                  offg.drawLine(xg,0,xg,crtgy);
                  offg.drawString(Integer.toString(i),xg,8);}
            for(j=2;j<=ni_depth;j+=2){
                  int yg=(int)(j*dy0);
                  offg.setColor(Color.black);
                  offg.drawLine(0,yg,crtgx,yg);
                  offg.drawString(Integer.toString(j),1,yg);}
            igrid=igrid+1;}
} // gridDraw
```

```
public void PseudoBoundary(){
        tf_command.setText("設定假想邊界端點");
        check=check+1;
        if(check==1) {xpr1=xc;ypr1=yc;
                JOptionPane.showMessageDialog(this, "按滑鼠設定
假想邊界第 2 端點"); }
        if(check==2) {xpr2=xc;ypr2=yc;
                JOptionPane.showMessageDialog(this, "按滑鼠設定
假想邊界第 3 端點"); }
        if(check==3) {xpr3=xc;ypr3=yc;
                JOptionPane.showMessageDialog(this, "按滑鼠設定
假想邊界第 4 端點"); }
        if(check==4) {xpr4=xc;ypr4=yc;}

        if(check==4)    {
            xtmp=xpr2-xpr1;
            ytmp=ypr2-ypr1;
            pseudoN1=(int)(Math.abs(ytemp)/dy0);
            for(i=1;i<=pseudoN1;i++){
                xc=(int)(xpr1+xtmp/(pseudoN1-1)*(i-1));
                yc=(int)(ypr1+ytmp/(pseudoN1-1)*(i-1));
                ibno1[i]=dki;
                px1[i]=xc;
                py1[i]=yc;
                AnyDepth(xc,yc,z_xy);
                pz1[i]=z_xy; }
            xtmp=xpr3-xpr2;
            ytmp=ypr3-ypr2;
            pseudoN2=(int)(Math.abs(xtemp)/dx0);

            for(i=1;i<=pseudoN2;i++){
                ii=i+pseudoN1;
                xc=(int)(xpr2+xtmp/pseudoN2*i);
                yc=(int)(ypr2+ytmp/pseudoN2*i);
                ibno1[ii]=dki;
                px1[ii]=xc;
                py1[ii]=yc;
                AnyDepth(xc,yc,z_xy);
```

```
                    pz1[ii]=z_xy;}
                xtmp=xpr4-xpr3;
                ytmp=ypr4-ypr3;
                pseudoN3=(int)(Math.abs(ytemp)/dy0);

                for(i=1;i<=pseudoN3;i++){
                    ii=i+pseudoN1+pseudoN2;
                    xc=(int)(xpr3+xtmp/pseudoN3*i);
                    yc=(int)(ypr3+ytmp/pseudoN3*i);
                    ibno1[ii]=dki;
                    px1[ii]=xc;
                    py1[ii]=yc;
                    AnyDepth(xc,yc,z_xy);
                    pz1[ii]=z_xy;}

            pseudoN=pseudoN1+pseudoN2+pseudoN3;}

            offg.drawLine(xpr1,ypr1,xpr2,ypr2);
            offg.drawLine(xpr2,ypr2,xpr3,ypr3);
            offg.drawLine(xpr3,ypr3,xpr4,ypr4);
            dki=pseudoN+1;
            for(i=1;i<=pseudoN;i++){offg.fillOval((int)px1[i],(int)py1[i],dot,dot);}
}       //PseudoBoundary

public void RightCoast(){
            tf_command.setText("右側海岸線");
            tf_node.setText(Integer.toString(dki));
            tf_z.setText(DoubleFormat.toString(z_xy,2));
            ibno1[dki]=dki;
            px1[dki]    =xc;
            py1[dki]    =yc;
            pz1[dki]    =0;
             offg.drawLine(xpr1,ypr1,xpr2,ypr2);
             offg.drawLine(xpr2,ypr2,xpr3,ypr3);
             offg.drawLine(xpr3,ypr3,xpr4,ypr4);
             for(i=1;i<=pseudoN;i++){offg.fillOval((int)px1[i],(int)py1[i],dot,dot);}
             for(i=pseudoN+1;i<dki;i++){
                    xx1=(int)px1[i-1];
```

```
                    yy1=(int)py1[i-1];
                    xx2=(int)px1[i];
                    yy2=(int)py1[i];
                    offg.fillOval(xx2,yy2,dot,dot);
                    offg.drawLine(xx1,yy1,xx2,yy2); }
                dki=dki+1;
}       //rightCoast

public void Basin(){
                tf_command.setText("港形");
                tf_node.setText(Integer.toString(dki));
                tf_z.setText(DoubleFormat.toString(z_xy,2));
                ibno1[dki]=dki;
                px1[dki]    =xc;
                py1[dki]    =yc;
                pz1[dki]    =z_xy;
                 offg.drawLine(xpr1,ypr1,xpr2,ypr2);
                 offg.drawLine(xpr2,ypr2,xpr3,ypr3);
                 offg.drawLine(xpr3,ypr3,xpr4,ypr4);
                 for(i=1;i<=pseudoN;i++){offg.fillOval((int)px1[i],(int)py1[i],dot,dot);}
                 for(i=pseudoN+1;i<dki;i++){
                    xx1=(int)px1[i-1];
                    yy1=(int)py1[i-1];
                    xx2=(int)px1[i];
                    yy2=(int)py1[i];
                    offg.fillOval(xx2,yy2,dot,dot);
                    offg.drawLine(xx1,yy1,xx2,yy2);}

                dki=dki+1;
}       //Basin

public void LeftCoast(){
                tf_command.setText("左側海岸線");
                tf_node.setText(Integer.toString(dki));
                tf_z.setText(DoubleFormat.toString(z_xy,2));
                ibno1[dki]=dki;
                px1[dki]    =xc;
                py1[dki]    =yc;
```

```
                pz1[dki]    =0;
                  offg.drawLine(xpr1,ypr1,xpr2,ypr2);
                  offg.drawLine(xpr2,ypr2,xpr3,ypr3);
                  offg.drawLine(xpr3,ypr3,xpr4,ypr4);
                  for(i=1;i<=pseudoN;i++){offg.fillOval((int)px1[i],(int)py1[i],dot,dot);}
                  for(i=pseudoN+1;i<dki;i++){
                        xx1=(int)px1[i-1];
                        yy1=(int)py1[i-1];
                        xx2=(int)px1[i];
                        yy2=(int)py1[i];
                        offg.fillOval(xx2,yy2,dot,dot);
                        offg.drawLine(xx1,yy1,xx2,yy2);
                        if(work==10) offg.drawLine(xx2,yy2,(int)px1[1],(int)py1[1]);
                        //if(work==10) islandStart[1]=dki+1;
                        if(work==10) End();}
                work=0;
                dki=dki+1;
}       //leftCoast

public void Island(){
              tf_command.setText("離島") ;
                tf_node.setText(Integer.toString(dki));
                tf_z.setText(DoubleFormat.toString(z_xy,2));
                px1[dki]=xc;
                py1[dki]=yc;
                pz1[dki]=z_xy;
                offg.drawLine(xpr1,ypr1,xpr2,ypr2);
                offg.drawLine(xpr2,ypr2,xpr3,ypr3);
                offg.drawLine(xpr3,ypr3,xpr4,ypr4);
                for(i=1;i<=pseudoN;i++){offg.fillOval((int)px1[i],(int)py1[i],dot,dot);}
                for(i=pseudoN+1;i<=tempN;i++){
                        xx1=(int)px1[i-1];
                        yy1=(int)py1[i-1];
                        xx2=(int)px1[i];
                        yy2=(int)py1[i];
                        offg.fillOval(xx2,yy2,dot,dot);
                        offg.drawLine(xx1,yy1,xx2,yy2);
                        if(i==tempN) offg.drawLine(xx2,yy2,(int)px1[1],(int)py1[1]);}
```

```
                if(nin==1){
                   if(dki==islandStart[1])    offg.fillOval(xc,yc,dot,dot);
                   for(i=islandStart[1]+1;i<=dki;i++){
                         xx1=(int)px1[i-1];
                         yy1=(int)py1[i-1];
                         xx2=(int)px1[i];
                         yy2=(int)py1[i];
                         offg.fillOval(xx2,yy2,dot,dot);
                         offg.drawLine(xx1,yy1,xx2,yy2);
                         if(work==10)
offg.drawLine(xx2,yy2,(int)px1[islandStart[1]],(int)py1[islandStart[1]]);
                         if(work==10) islandEnd[1]=dki;
                         if(work==10) End();}
                }
                work=0;
                if(nin>1){
                   for(ii=2;ii<=nin;ii++){
                         islandStart[ii]=islandEnd[ii-1]+1;
                        if(dki==islandStart[ii])    offg.fillOval(xc,yc,dot,dot);
                         for(i=islandStart[ii]+1;i<=dki;i++){
                             xx1=(int)px1[i-1];
                             yy1=(int)py1[i-1];
                             xx2=(int)px1[i];
                             yy2=(int)py1[i];
                             offg.fillOval(xx2,yy2,dot,dot);
                             offg.drawLine(xx1,yy1,xx2,yy2);
                             if(work==10)
offg.drawLine(xx2,yy2,(int)px1[islandStart[ii]],(int)py1[islandStart[ii]]);
                             //if(work==10) islandEnd[ii]=dki;
                             if(work==10) End();}
                         work=0;}
                }
                  dki=dki+1;
}        // Island

public void InnerNode(){
             tf_command.setText("內部網格節點") ;
```

```
            dx0=(crtgx/(double)nj_depth);
            dy0=(crtgy/(double)ni_depth);
            innerNode=1;
            for(i=1;i<=nj_depth-1;i+=gridinterval){  xc=(int)(i*dx0);
            for(j=1;j<=ni_depth-1;j+=gridinterval){ yc=(int)(j*dy0);

               if(clockwise==1){
                  xl=xpr1+(int)((xpr2-xpr1)/(double)(ypr2-ypr1)*(yc-ypr1));
                  xr=xpr4+(int)((xpr3-xpr4)/(double)(ypr3-ypr4)*(yc-ypr4));

                  if(yc<=ypr2)                  continue;
                  if(xc<=xl && yc<=ypr1)   continue;
                  if(xc>=xr && yc<=ypr4)   continue;
                  if(gridDepth[i][j]>depthIndex)     continue;
                  px1[dki]=xc;
                  py1[dki]=yc;
                  pz1[dki]= gridDepth[i][j];}
               if(clockwise==2){
                  xr=xpr1+(int)((xpr2-xpr1)/(double)(ypr2-ypr1)*(yc-ypr1));
                  xl=xpr4+(int)((xpr3-xpr4)/(double)(ypr3-ypr4)*(yc-ypr4));

                  if(yc<=ypr2)                  continue;
                  if(xc<=xl && yc<=ypr4)   continue;
                  if(xc>=xr && yc<=ypr1)   continue;
                  if(gridDepth[i][j]>depthIndex)     continue;
                  px1[dki]=xc;
                  py1[dki]=yc;
                  pz1[dki]= gridDepth[i][j];}
                  dki=dki+1;
                  innerNode=innerNode+1;
                  }}

} //InnerNode
public void InnerPoint(){
                tf_command.setText("內部加強節點") ;
                tf_node.setText(Integer.toString(dki));
                tf_z.setText(DoubleFormat.toString(z_xy,2));
```

```
                px1[dki]    =xc;
                py1[dki]    =yc;
                pz1[dki]    =z_xy;
                innerPoint=1;
                offg.drawLine(xpr1,ypr1,xpr2,ypr2);
                offg.drawLine(xpr2,ypr2,xpr3,ypr3);
                offg.drawLine(xpr3,ypr3,xpr4,ypr4);

                for(i=1;i<=pseudoN;i++){offg.fillOval((int)px1[i],(int)py1[i],dot,dot);}
                for(i=pseudoN+1;i<=tempN;i++){
                        xx1=(int)px1[i-1];
                        yy1=(int)py1[i-1];
                        xx2=(int)px1[i];
                        yy2=(int)py1[i];
                        offg.fillOval(xx2,yy2,dot,dot);
                        offg.drawLine(xx1,yy1,xx2,yy2);
                        if(i==tempN) offg.drawLine(xx2,yy2,(int)px1[1],(int)py1[1]);}

              if(nin==1){
                 if(dki==islandStart[1])    offg.fillOval(xc,yc,dot,dot);
                 for(i=islandStart[1]+1;i<=dki;i++){
                        xx1=(int)px1[i-1];
                        yy1=(int)py1[i-1];
                        xx2=(int)px1[i];
                        yy2=(int)py1[i];
                        offg.fillOval(xx2,yy2,dot,dot);
                        offg.drawLine(xx1,yy1,xx2,yy2);
                        if(work==10)
offg.drawLine(xx2,yy2,(int)px1[islandStart[1]],(int)py1[islandStart[1]]);
                        if(work==10) islandEnd[1]=dki;}
              }
              work=0;
              if(nin>1){
                 for(ii=2;ii<=nin;ii++){
                        islandStart[ii]=islandEnd[ii-1]+1;
                       if(dki==islandStart[ii])    offg.fillOval(xc,yc,dot,dot);
                       for(i=islandStart[ii]+1;i<=dki;i++){
                           xx1=(int)px1[i-1];
```

```
                        yy1=(int)py1[i-1];
                        xx2=(int)px1[i];
                        yy2=(int)py1[i];
                        offg.fillOval(xx2,yy2,dot,dot);
                        offg.drawLine(xx1,yy1,xx2,yy2);
                        if(work==10)
offg.drawLine(xx2,yy2,(int)px1[islandStart[ii]],(int)py1[islandStart[ii]]);
                        if(work==10) islandEnd[ii]=dki;}
                    work=0;}
                }
                  offg.fillOval(xc,yc,dot,dot);

                  dki=dki+1;
                  innerPoint=innerPoint+1;
} //InnerPoint

public void TrianglePlot(int node,int nelm,double []px,double []py,int [][]mtj){

          xmin = px[1]; xmax = xmin;
          ymin = py[1]; ymax = ymin;
          for(i=1;i<=node;i++){
                if(px[i]<xmin) xmin=px[i];
                if(xmax<px[i]) xmax=px[i];
                if(py[i]<ymin) ymin=py[i];
                if(ymax<py[i]) ymax=py[i];}

          x_plot=(int)(xmax-xmin);
          y_plot=(int)(ymax-ymin);


          if(x_plot<y_plot ) ds=(double)lengthh/y_plot;
          if(x_plot>=y_plot) ds=(double)lengthh/x_plot;

          hsize=((int)(ds*x_plot)+100)*kpt;
          vsize=((int)(ds*y_plot)+100)*kpt;

          kxi=50*kpt; kxf=hsize-50*kpt;
          kyi=50*kpt; kyf=vsize-50*kpt;
```

```
 for(ne=1;ne<=nelm;ne++){
      nod1=mtj[ne][1];    nod2=mtj[ne][2];        nod3=mtj[ne][3];
      xx=(int)px[nod1]; kxx1=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
      yy=(int)py[nod1]; kyy1=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
      xx=(int)px[nod2]; kxx2=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
      yy=(int)py[nod2]; kyy2=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
      xx=(int)px[nod3]; kxx3=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
      yy=(int)py[nod3]; kyy3=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));

      offg.drawLine(kxx1,kyy1,kxx2,kyy2);
      offg.drawLine(kxx2,kyy2,kxx3,kyy3);
      offg.drawLine(kxx3,kyy3,kxx1,kyy1); }

   for(ne=1;ne<=nelm;ne++){
      nod1=mtj[ne][1];    nod2=mtj[ne][2];        nod3=mtj[ne][3];
      xx=(int)px[nod1]; kxx1=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
      yy=(int)py[nod1]; kyy1=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
      xx=(int)px[nod2]; kxx2=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
      yy=(int)py[nod2]; kyy2=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
      xx=(int)px[nod3]; kxx3=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
      yy=(int)py[nod3]; kyy3=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));

      offg.drawLine(kxx1,lengthh+10+kyy1,kxx2,lengthh+10+kyy2);
      offg.drawLine(kxx2,lengthh+10+kyy2,kxx3,lengthh+10+kyy3);

      offg.drawLine(kxx3,lengthh+10+kyy3,kxx1,lengthh+10+kyy1); }

        for(i=1;i<=node;i++){
            xx=(int)px[i]; kxx1=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-
xmin));
            yy=(int)py[i]; kyy1=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-
ymin));
            offg.drawString(Integer.toString(i),kxx1,lengthh+10+kyy1);}

   for(ne=1;ne<=nelm;ne++){
      nod1=mtj[ne][1];    nod2=mtj[ne][2];        nod3=mtj[ne][3];
      xx=(int)px[nod1]; kxx1=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
```

```
                yy=(int)py[nod1]; kyy1=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                xx=(int)px[nod2]; kxx2=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod2]; kyy2=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                xx=(int)px[nod3]; kxx3=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod3]; kyy3=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));


offg.drawLine(lengthh+10+kxx1,lengthh+10+kyy1,lengthh+10+kxx2,lengthh+10+kyy2
);

offg.drawLine(lengthh+10+kxx2,lengthh+10+kyy2,lengthh+10+kxx3,lengthh+10+kyy3
);

offg.drawLine(lengthh+10+kxx3,lengthh+10+kyy3,lengthh+10+kxx1,lengthh+10+kyy1
); }
                    for(ne=1;ne<=nelm;ne++){
                        nod1=mtj[ne][1]; nod2=mtj[ne][2]; nod3=mtj[ne][3];
                        xx=(int)((px[nod1]+px[nod2]+px[nod3])/3.) ;
kxx1=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                        yy=(int)((py[nod1]+py[nod2]+py[nod3])/3.) ; kyy1=kyf-
(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));

offg.drawString(Integer.toString(ne),lengthh+10+kxx1,lengthh+10+kyy1); }

            for(ne=1;ne<=nelm;ne++){
                    i=mtj[ne][1];
                    j=mtj[ne][2];
                    k=mtj[ne][3];
                    area1[ne]=0.5*((px[k]-px[j])*py[i]+(px[i]-px[k])*py[j]+(px[j]-
px[i])*py[k]);}

          amean=0.0;
          amax =area1[1];
          amin =area1[1];

            for(ne=1;ne<=nelm;ne++){
                amean = amean + area1[ne];
                if(amax < area1[ne]) amax = area1[ne];
```

```
                if(area1[ne] < amin) amin = area1[ne];}

            amean=amean/(double)(nelm);

}     //plot

public void SquarePlot(int node,int nelm,double []px,double []py,int [][]mtj){

            xmin = px[1] ; xmax = xmin;
            ymin = py[1] ; ymax = ymin;
            for(i = 1;i<=node;i++){
                if(px[i]<xmin) xmin = px[i];
                if(xmax<px[i]) xmax = px[i];
                if(py[i]<ymin) ymin = py[i];
                if(ymax<py[i]) ymax = py[i]; }

            x_plot=(int)(xmax-xmin);
            y_plot=(int)(ymax-ymin);

            if(x_plot<y_plot ) ds=(double)lengthh/y_plot;
            if(x_plot>=y_plot) ds=(double)lengthh/x_plot;

            hsize=((int)(ds*x_plot)+100)*kpt;
            vsize=((int)(ds*y_plot)+100)*kpt;

            kxi=50*kpt; kxf=hsize-50*kpt;
            kyi=50*kpt; kyf=vsize-50*kpt;

              for(ne=1;ne<=nelm;ne++){
                nod1=mtj[ne][1];      nod2=mtj[ne][2];          nod3=mtj[ne][3];
nod4=mtj[ne][4];
                xx=(int)px[nod1]; kxx1=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod1]; kyy1=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                xx=(int)px[nod2]; kxx2=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod2]; kyy2=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                xx=(int)px[nod3]; kxx3=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod3]; kyy3=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                xx=(int)px[nod4]; kxx4=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
```

```
                yy=(int)py[nod4]; kyy4=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));

                offg.drawLine(kxx1,kyy1,kxx2,kyy2);
                offg.drawLine(kxx2,kyy2,kxx3,kyy3);
                offg.drawLine(kxx3,kyy3,kxx4,kyy4);
                offg.drawLine(kxx4,kyy4,kxx1,kyy1);            }

          for(ne=1;ne<=nelm;ne++){
                nod1=mtj[ne][1];     nod2=mtj[ne][2];          nod3=mtj[ne][3];
nod4=mtj[ne][4];
                xx=(int)px[nod1]; kxx1=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod1]; kyy1=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                xx=(int)px[nod2]; kxx2=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod2]; kyy2=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                xx=(int)px[nod3]; kxx3=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod3]; kyy3=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                xx=(int)px[nod4]; kxx4=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod4]; kyy4=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));

                offg.drawLine(kxx1,lengthh+10+kyy1,kxx2,lengthh+10+kyy2);
                offg.drawLine(kxx2,lengthh+10+kyy2,kxx3,lengthh+10+kyy3);
                offg.drawLine(kxx3,lengthh+10+kyy3,kxx4,lengthh+10+kyy4);

                offg.drawLine(kxx1,lengthh+10+kyy1,kxx1,lengthh+10+kyy1); }

                for(i=1;i<=node;i++){
                     xx=(int)px[i]; kxx1=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                     yy=(int)py[i]; kyy1=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                     offg.drawString(Integer.toString(i),kxx1,lengthh+10+kyy1);}

          for(ne=1;ne<=nelm;ne++){
                nod1=mtj[ne][1];     nod2=mtj[ne][2];          nod3=mtj[ne][3];
nod4=mtj[ne][4];
                xx=(int)px[nod1]; kxx1=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod1]; kyy1=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                xx=(int)px[nod2]; kxx2=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod2]; kyy2=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                xx=(int)px[nod3]; kxx3=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
```

```
                yy=(int)py[nod3]; kyy3=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));
                xx=(int)px[nod4]; kxx4=kxi+(int)((xx-xmin)*(kxf-kxi)/(xmax-xmin));
                yy=(int)py[nod4]; kyy4=kyf-(int)((yy-ymin)*(kyf-kyi)/(ymax-ymin));


offg.drawLine(lengthh+10+kxx1,lengthh+10+kyy1,lengthh+10+kxx2,lengthh+10+kyy2
);

offg.drawLine(lengthh+10+kxx2,lengthh+10+kyy2,lengthh+10+kxx3,lengthh+10+kyy3
);

offg.drawLine(lengthh+10+kxx3,lengthh+10+kyy3,lengthh+10+kxx4,lengthh+10+kyy4
);

offg.drawLine(lengthh+10+kxx4,lengthh+10+kyy4,lengthh+10+kxx1,lengthh+10+kyy1
); }

                for(ne=1;ne<=nelm;ne++){
                    nod1=mtj[ne][1]; nod2=mtj[ne][2]; nod3=mtj[ne][3];
                    xx=(int)((px[nod1]+px[nod2]+px[nod3])/3.) ; kxx1=kxi+(int)((xx-
xmin)*(kxf-kxi)/(xmax-xmin));
                    yy=(int)((py[nod1]+py[nod2]+py[nod3])/3.) ; kyy1=kyf-(int)((yy-
ymin)*(kyf-kyi)/(ymax-ymin));

offg.drawString(Integer.toString(ne),lengthh+10+kxx1,lengthh+10+kyy1); }

            double []tarea=new double[nelm];
            double elarea1,elarea2,amean,amax,amin;
            double fact0 = 1.0e-30;
            int iarea= 0;

            for(ne = 1;ne<=nelm;ne++){
                nod1 = mtj[ne][1];    nod2 = mtj[ne][2]; nod3 = mtj[ne][3];
                elarea1=.5*((px[nod3]-px[nod2])*py[nod1]+(px[nod1]-
px[nod3])*py[nod2]+(px[nod2]-px[nod1])*py[nod3]);
                if(elarea1 < fact0) iarea = iarea + 1;

                nod1 = mtj[ne][1] ; nod2 = mtj[ne][3] ; nod3 = mtj[ne][4];
```

```
                elarea2=.5*((px[nod3]-px[nod2])*py[nod1]+(px[nod1]-
px[nod3])*py[nod2]+(px[nod2]-px[nod1])*py[nod3]);
                if(elarea2 < fact0) iarea = iarea + 1;
                tarea[ne]= elarea1 + elarea2; }

          amean= 0.0 ;
          amax = tarea[1] ;
          amin = tarea[1];
          for(ne = 1;ne<=nelm;ne++){
                amean = amean + tarea[ne];
                if(amax<tarea[ne]) amax = tarea[ne];
                if(tarea[ne]<amin) amin = tarea[ne];
          }
          amean = amean /(double)nelm;

   } //SquarePlot


} //Figure
public void normal(int node,int nelm,double []px,double []py,double []pz,int
[][]mtj,double []xn,double []yn,double []zn,double []area){
      double r1,r2,r3,r4,r5,r6,r,t1,t2,t3,t4,t5,t6,t;
      for(i=1;i<=nelm;i++){
          r1=px[mtj[i][3]]-px[mtj[i][1]];
          r2=py[mtj[i][3]]-py[mtj[i][1]];
          r3=pz[mtj[i][3]]-pz[mtj[i][1]];
          r4=px[mtj[i][4]]-px[mtj[i][2]];
          r5=py[mtj[i][4]]-py[mtj[i][2]];
          r6=pz[mtj[i][4]]-pz[mtj[i][2]];

          r=Math.sqrt((r5*r3-r6*r2)*(r5*r3-r6*r2)+(r6*r1-r4*r3)*(r6*r1-
r4*r3)+(r4*r2-r5*r1)*(r4*r2-r5*r1));

          xn[i]=(r5*r3-r6*r2)/r;
          yn[i]=(r6*r1-r4*r3)/r;
          zn[i]=(r4*r2-r5*r1)/r;

          t1=px[mtj[i][3]]-px[mtj[i][1]];
```

```
            t2=py[mtj[i][3]]-py[mtj[i][1]];
            t3=pz[mtj[i][3]]-pz[mtj[i][1]];
            t4=px[mtj[i][4]]-px[mtj[i][1]];
            t5=py[mtj[i][4]]-py[mtj[i][1]];
            t6=pz[mtj[i][4]]-pz[mtj[i][1]];

            r=.5*Math.sqrt((t2*r3-t3*r2)*(t2*r3-t3*r2)+(t3*r1-t1*r3)*(t3*r1-
t1*r3)+(t1*r2-t2*r1)*(t1*r2-t2*r1));
            t=.5*Math.sqrt((r2*t6-r3*t5)*(r2*t6-r3*t5)+(r3*t4-r1*t6)*(r3*t4-
r1*t6)+(r1*t5-r2*t4)*(r1*t5-r2*t4));
            area[i]=r+t;}
}// normal

public void LeftBoundary(){
      tf_command.setText("左側邊界面");

      int n1=pseudoN1;
      node3=n1*3+1;
      nelm3=2*n1-1;

         px3=new double[node3];
         py3=new double[node3];
         pz3=new double[node3];
        mtj3=new int[nelm3][4];
        jac3=new int[nelm3][4];
        nei3=new int[nelm3][kcm];
        jnb3=new int[nelm3];
         xn3=new double[nelm3];
         yn3=new double[nelm3];
         zn3=new double[nelm3];
      area3=new double[nelm3];

      for(i=1;i<=node3;i++){for(j=1;j<=kcm;j++){nei3[i][j]=0;}}

      px3[1] =px1[1];py3[1]=py1[1];pz3[1]=pz1[1];

      for(i=2;i<=n1;i++) {
      for(j=1;j<=3;j++) {
```

```
            ii=(n1-2+1)*(j-1)+i;
            if(j==1){px3[ii]=px1[i];py3[ii]=py1[i];pz3[ii]=0             ;}
            if(j==2){px3[ii]=px1[i];py3[ii]=py1[i];pz3[ii]=pz1[i]/2;}
            if(j==3){px3[ii]=px1[i];py3[ii]=py1[i];pz3[ii]=pz1[i]    ;}
      }}

      px3[3*n1-1]=(px3[1]+px3[2])/2;                  py3[3*n1-1]=(py3[1]+py3[2])/2;
pz3[3*n1-1]=0;

px3[3*n1    ]=(px3[1]+px3[2]+px3[2*n1])/3;py3[3*n1    ]=(py3[1]+py3[2]+py3[2*n1])/
3;pz3[3*n1    ]=(pz3[1]+pz3[2]+pz3[2*n1])/3;
      px3[3*n1+1]=(px3[1]+px3[2*n1])/2;
py3[3*n1+1]=(py3[1]+py3[2*n1])/2;             pz3[3*n1+1]=(pz3[1]+pz3[2*n1])/2;

      mtj3[1][1]=3*n1+1; mtj3[1][2]=3*n1; mtj3[1][3]=3*n1-1; mtj3[1][4]=1;
      mtj3[2][1]=3*n1;      mtj3[2][2]=n1+1; mtj3[2][3]=2;           mtj3[2][4]=3*n1-1;
      mtj3[3][1]=3*n1+1; mtj3[3][2]=2*n1; mtj3[3][3]=n1+1;      mtj3[3][4]=3*n1;
  //ok
      for(i=4;i<=n1+1;i++)       {mtj3[i][1]=n1+i-3; mtj3[i][2]=n1+i-2; mtj3[i][3]=i-1;
mtj3[i][4]=i-2;}
      for(i=n1+2;i<=nelm3;i++){mtj3[i][1]=n1+i-2; mtj3[i][2]=n1+i-1; mtj3[i][3]=i;
mtj3[i][4]=i-1;}

      jnb3[1]=1;
      jnb3[3*n1-1]=2;
      for(i=2;i<=n1-1;i++) {jnb3[i]=2;}
      jnb3[n1]=1;

      jnb3[3*n1]=3;
      for(i=n1+1;i<2*n1-1;i++) {jnb3[i]=4;}
      jnb3[2*n1-1]=2;

      jnb3[3*n1+1]=2;
      for(i=n1*2;i<3*n1-2;i++) {jnb3[i]=2;}
      jnb3[3*n1-2]=1;

      nei3[1][1]=1;
      nei3[3*n1-1][1]=1;    nei3[3*n1-1][2]=2;
```

```
        nei3[2][1]=4;           nei3[2][2]       =2;
        nei3[3*n1][1]    =1;    nei3[3*n1][2]    =2;    nei3[3*n1][3]=3;
        nei3[n1+1][1]=4;        nei3[n1+1][2]=n1+2;    nei3[n1+1][3]=3;
nei3[n1+1][4]=2;
        nei3[3*n1+1][1]=1;    nei3[3*n1+1][2]=3;
        nei3[2*n1][1]=n1+2;    nei3[2*n1][2]=3;

        for(i=3;i<=n1;i++) {
        for(j=1;j<=3;j++) {
              ii=(n1-3+1)*(j-1)+i;
              if(j==1){nei3[ii][1]=i+1;    nei3[ii][2]=i+2;}
              if(j==2){nei3[ii][1]=i+1;    nei3[ii][2]=n1+i-1; nei3[ii][3]=n1+i;
nei3[ii][4]=i+2;}
              if(j==3){nei3[ii][1]=n1+i-1;nei3[ii][2]=n1+i;}
        }}
        nei3[n1][1]=n1+1;
        nei3[2*n1-1][1]=n1+1;    nei3[1*n1-1][2]=2*n1-1;
        nei3[3*n1-2][1]=2*n1-1;

        normal(node3,nelm3,px3,py3,pz3,mtj3,xn3,yn3,zn3,area3);

}

public void MiddleBoundary(){
        tf_command.setText("中間邊界面");
        int n2=pseudoN2;
        int n=n2+1;
        node2=n*3;
        nelm2=2*n2;

           px2=new double[node2];
           py2=new double[node2];
           pz2=new double[node2];
         mtj2=new int[nelm2][4];
         jac2=new int[nelm2][4];
         nei2=new int[nelm2][kcm];
         jnb2=new int[nelm2];
           xn2=new double[nelm2];
```

```
    yn2=new double[nelm2];
    zn2=new double[nelm2];
  area2=new double[nelm2];

  for(i=1;i<=node2;i++){for(j=1;j<=kcm;j++){nei2[i][j]=0;}}

  for(i=1;i<=n;i++) {
      iii=pseudoN1-1+i;
  for(j=1;j<=3;j++) {
      ii=n*(j-1)+i;
      if(j==1){px2[ii]=px1[iii];py2[ii]=py1[iii];pz2[ii]=0                ;}
      if(j==2){px2[ii]=px1[iii];py2[ii]=py1[iii];pz2[ii]=pz1[iii]/2;}
      if(j==3){px2[ii]=px1[iii];py2[ii]=py1[iii];pz2[ii]=pz1[iii]    ;}
  }}

  for(i=1;    i<=n2;i++)    {mtj2[i][1]=n2+1+i; mtj2[i][2]=n2+2+i; mtj2[i][3]=i+1;
mtj2[i][4]=i;}
  for(i=n2+1;i<=nelm2;i++){mtj2[i][1]=n2+2+i; mtj2[i][2]=n2+3+i; mtj2[i][3]=i+2;
mtj2[i][4]=i+1;}

  jnb2[1]=1;     jnb2[n2+2]=2; jnb2[2*n2+3]=1;

  for(i=2;      i<=n-1;   i++) {jnb2[i]=2;}
  for(i=n+2;    i<=2*n-1;i++) {jnb2[i]=4;}
  for(i=2*n+2;i<=3*n-1;i++) {jnb2[i]=2;}

  jnb2[n]=1;     jnb2[n*2]=2;   jnb2[3*n]=1;

  nei2[1][1]      =1;
  nei2[n+1][1]   =1;      nei2[n+1][2]=n;
  nei2[2*n+1][1]=n;

  for(i=2;i<n; i++) {
  for(j=1;j<=3;j++) {
      ii=n*(j-1)+i;
      if(j==1){nei2[ii][1]=i-1;    nei2[ii][2]=i;}
      if(j==2){nei2[ii][1]=i-1;    nei2[ii][2]=n+i-1; nei2[ii][3]=n+i;    nei2[ii][4]=i;}
      if(j==3){nei2[ii][1]=i-1;    nei2[ii][2]=i;}
```

```
        }}
        nei2[n][1]      =n-1;
        nei2[2*n][1]    =n-1;       nei2[2*n][2]=2*(n-1);
        nei2[2*n+1][1]=3*(n-1);

        normal(node2,nelm2,px2,py2,pz2,mtj2,xn2,yn2,zn2,area2);

}

public void RightBoundary(){
        tf_command.setText("右側邊界面");
        int n3=pseudoN3;
        int n=n3+1;
        node4=3*n+1;
        nelm4=2*n-1;

           px4=new double[node4];
           py4=new double[node4];
           pz4=new double[node4];
         mtj4=new int[nelm4][4];
         jac4=new int[nelm4][4];
         nei4=new int[nelm4][kcm];
         jnb4=new int[nelm4];
           xn4=new double[nelm4];
           yn4=new double[nelm4];
           zn4=new double[nelm4];
        area4=new double[nelm4];

        for(i=1;i<=node4;i++){for(j=1;j<=kcm;j++){nei4[i][j]=0;}}

        px4[1]=px1[pseudoN];py4[pseudoN]=py1[pseudoN];pz4[pseudoN]=pz1[1];

        for(i=2;i<=n;i++) {
                iii=pseudoN+1-i;
        for(j=1;j<=3;j++) {
                ii=(n-2)*(j-1)+i;
                if(j==1){px4[ii]=px1[iii];py4[ii]=py1[iii];pz4[ii]=0                ;}
                if(j==2){px4[ii]=px1[iii];py4[ii]=py1[iii];pz4[ii]=pz1[iii]/2;}
```

```
            if(j==3){px4[ii]=px1[iii];py4[ii]=py1[iii];pz4[ii]=pz1[iii]    ;}
        }}

        px4[3*n-1]=(px4[1]+px4[2])/2;                py4[3*n-1]=(py4[1]+py4[2])/2;
pz4[3*n-1]=0;
        px4[3*n]    =(px4[1]+px4[2]+px4[2*n])/3;py4[3*n]
=(py4[1]+py4[2]+py4[2*n])/3;pz4[3*n]=(pz4[1]+pz4[2]+pz4[2*n])/3;
        px4[3*n+1]=(px4[1]+px4[2*n])/2;             py4[3*n+1]=(py4[1]+py4[2*n])/2;
pz4[3*n+1]=(pz4[1]+pz4[2*n])/2;

        mtj4[1][1]=3*n+1; mtj4[1][2]=3*n; mtj4[1][3]=3*n-1; mtj4[1][4]=1;
        mtj4[2][1]=3*n;      mtj4[2][2]=n+1; mtj4[2][3]=2;          mtj4[2][4]=3*n-1;
        mtj4[3][1]=3*n+1; mtj4[3][2]=2*n; mtj4[3][3]=n+1;      mtj4[3][4]=3*n;

        for(i=4;i<=n+1;i++)        {mtj4[i][1]=n+i-3; mtj4[i][2]=n+i-2; mtj4[i][3]=i-1;
mtj4[i][4]=i-2;}
        for(i=n+2;i<=nelm3;i++){mtj4[i][1]=i+n-2; mtj4[i][2]=i+n-1; mtj4[i][3]=i;
mtj4[i][4]=i-1;}

        jnb4[1]=1;
        jnb4[3*n-1]=2;
        for(i=2;i<=n-1;i++) {jnb4[i]=2;}
        jnb4[n]=1;
        jnb4[3*n]=3;
        for(i=n+1;i<2*n-1;i++) {jnb4[i]=4;}
        jnb4[2*n-1]=1;
        jnb4[3*n+1]=2;
        for(i=n*2;i<3*n-3;i++) {jnb4[i]=2;}
        jnb4[3*n-2]=1;

        nei4[1][1]=1;
        nei4[3*n-1][1]=1;    nei4[3*n-1][2]=2;
        nei4[2][1]=4;           nei4[2][2]       =2;
        nei4[3*n][1]    =1;    nei4[3*n][2]    =2;    nei4[3*n][3]=3;
        nei4[n+1][1]=4;        nei4[n+1][2]=n+2;    nei4[n+1][3]=3;    nei4[n+1][4]=2;
        nei4[3*n+1][1]=1;    nei4[3*n+1][2]=3;
        nei4[2*n][1]=n+2;    nei4[2*n][2]=3;
```

```
        for(i=3;i<n;i++) {
        for(j=1;j<=3;j++) {
              ii=(n-3+1)*(j-1)+i;
              if(j==1){nei4[ii][1]=i+1;    nei4[ii][2]=i+2;}
              if(j==2){nei4[ii][1]=i+1;    nei4[ii][2]=n+i-1; nei4[ii][3]=n+i;    nei4[ii][4]=i+2;}
              if(j==3){nei4[ii][1]=n+i-1;nei4[ii][2]=n+i;}
        }}
        nei4[n][1]=n+1;
        nei4[2*n-1][1]=n+1;    nei4[1*n-1][2]=2*n-1;
        nei4[3*n-2][1]=2*n-1;

        normal(node4,nelm4,px4,py4,pz4,mtj4,xn4,yn4,zn4,area4);

}

public void Quay(){
        tf_command.setText("岸壁面");

        int nq=quayNum;
        int n=quayNum-1;
        int tmp=pseudoN+rightcoastNum;

        node5=quayNum*3+2;
        nelm5=quayNum*2;

           px5=new double[node5];
           py5=new double[node5];
           pz5=new double[node5];
         mtj5=new int[nelm5][4];
         jac5=new int[nelm5][4];
         nei5=new int[nelm5][kcm];
         jnb5=new int[nelm5];
           xn5=new double[nelm5];
           yn5=new double[nelm5];
           zn5=new double[nelm5];
        area5=new double[nelm5];

        for(i=1;i<=node5;i++){for(j=1;j<=kcm;j++){nei5[i][j]=0;}}
```

```
px5[1] =px1[tmp+1];py5[1]=py1[tmp+1];pz5[1]=pz1[tmp+1];
px5[n] =px1[tmp+n];py5[n]=py1[tmp+n];pz5[tmp+n]=pz1[1];

for(i=2;i<=n;i++) {
for(j=1;j<=3;j++) {
    ii=(n-2+1)*(j-1)+i;
    if(j==1){px5[ii]=px1[i];py5[ii]=py1[i];pz5[ii]=0          ;}
    if(j==2){px5[ii+1]=px1[i];py5[ii+1]=py1[i];pz5[ii+1]=pz1[i]/2;}
    if(j==3){px5[ii+1]=px1[i];py5[ii+1]=py1[i];pz5[ii+1]=pz1[i]  ;}
}}

px5[3*n-1]=(px5[1]+px5[2])/2;              py5[3*n-1]=(py5[1]+py5[2])/2;
pz5[3*n-1]=0;
px5[3*n]  =(px5[1]+px5[2]+px5[2*n])/3;py5[3*n]
=(py5[1]+py5[2]+py5[2*n])/3;pz5[3*n]=(pz5[1]+pz5[2]+pz5[2*n])/3;
px5[3*n+1]=(px5[1]+px5[2*n])/2;            py5[3*n+1]=(py5[1]+py5[2*n])/2;
pz5[3*n+1]=(pz5[1]+pz5[2*n])/2;

mtj5[1][1]=3*nq-1; mtj5[1][2]=3*nq-2; mtj5[1][3]=3*nq-3; mtj5[1][4]=1;
mtj5[2][1]=3*nq-2; mtj5[2][2]=nq+1;    mtj5[2][3]=2;          mtj5[2][4]=3*nq-3;
mtj5[3][1]=3*nq-1; mtj5[3][2]=2*nq-1; mtj5[3][3]=nq+1;    mtj5[3][4]=3*nq-2;

px5[3*n+2]=(px5[nq+n]+px5[nq+n])/2;
py5[3*n+2]=(py5[nq+n]+py5[nq+n])/2;            pz5[3*n+2]=0;
px5[3*n+3]=(px5[nq]+px5[n]+px5[3*nq-
4])/3;py5[3*n+3]=(py5[nq]+py5[n]+py5[3*nq-
4])/3;pz5[3*n+3]=(pz5[nq]+pz5[n]+pz5[3*nq-4])/3;
px5[3*n+4]=(px5[nq]+px5[3*nq-4])/2;            py5[3*n+4]=(py5[nq]+py5[3*nq-
4])/2;            pz5[3*n+4]=(pz5[nq]+pz5[3*nq-4])/2;

mtj5[2*nq-2][1]=3*nq+2; mtj5[2*nq-2][2]=nq;        mtj5[2*nq-2][3]=3*nq;
mtj5[2*nq-2][4]=3*nq+1;
mtj5[2*nq-1][1]=2*nq-2; mtj5[2*nq-1][2]=3*nq+1; mtj5[2*nq-1][3]=3*nq;
mtj5[2*nq-1][4]=nq-1;
mtj5[2*nq  ][1]=3*nq-4; mtj5[2*nq  ][2]=3*nq+2; mtj5[2*nq  ][3]=3*nq+1;
mtj5[2*nq  ][4]=2*nq-2;
```

```
    for(i=4;i<=n+1;i++)          {mtj5[i][1]=n+i-3; mtj5[i][2]=n+i-2; mtj5[i][3]=i-1;
mtj5[i][4]=i-2;}
    for(i=n+2;i<=nelm5-3;i++){mtj5[i][1]=n+i-2; mtj5[i][2]=n+i-1; mtj5[i][3]=i;
mtj5[i][4]=i-1;}

    jnb5[1]=1;
    jnb5[3*n-1]=2;
    for(i=2;i<=n-1;i++) {jnb5[i]=2;}
    jnb5[n]=1;
    jnb5[3*n]=3;
    for(i=n+1;i<2*n-1;i++) {jnb5[i]=4;}
    jnb5[2*n-1]=1;
    jnb5[3*n+1]=2;
    for(i=n*2;i<3*n-3;i++) {jnb5[i]=2;}
    jnb5[3*n-2]=1;

    nei5[1][1]=1;
    nei5[3*n-1][1]=1;   nei5[3*n-1][2]=2;
    nei5[2][1]=4;          nei5[2][2]      =2;
    nei5[3*n][1]   =1;   nei5[3*n][2]   =2;   nei5[3*n][3]=3;
    nei5[n+1][1]=4;      nei5[n+1][2]=n+2;   nei5[n+1][3]=3;   nei5[n+1][4]=2;
    nei5[3*n+1][1]=1;   nei5[3*n+1][2]=3;
    nei5[2*n][1]=n+2;   nei5[2*n][2]=3;

    for(i=3;i<n;i++) {
    for(j=1;j<=3;j++) {
        ii=(n-3+1)*(j-1)+i;
        if(j==1){nei5[ii][1]=i+1;   nei5[ii][2]=i+2;}
        if(j==2){nei5[ii][1]=i+1;   nei5[ii][2]=n+i-1; nei5[ii][3]=n+i;   nei5[ii][4]=i+2;}
        if(j==3){nei5[ii][1]=n+i-1;nei5[ii][2]=n+i;}
    }}

    nei5[nq][1]=2*nq-2;
    nei5[3*nq][1]=2*nq-1;     nei5[3*nq][2]=2*nq-2;
    nei5[n][1]=nq;                nei5[n][2]=2*nq-1;
    nei5[3*nq+1][1]=2*nq-1; nei5[3*nq+1][2]=2*nq;     nei5[3*nq+1][3]=2*nq-2;

    nei5[2*nq-2][1]=nq;         nei5[2*nq-2][2]=2*nq-3; nei5[2*nq-2][3]=2*nq;
```

```
nei5[2*nq-2][4]=2*nq-1;
        nei5[3*nq-4][1]=2*nq-3; nei5[3*nq-4][2]=2*nq;
        nei5[3*nq+2][1]=2*nq;      nei5[3*nq+2][2]=2*nq-2;

        normal(node5,nelm5,px5,py5,pz5,mtj5,xn5,yn5,zn5,area5);

}//quay
public void Seabed(){
        tf_command.setText("海底面");

        node6=node1;
        nelm6=nelm1;
           px6=new double[node6];
           py6=new double[node6];
           pz6=new double[node6];
         mtj6=new int[nelm6][4];
         jac6=new int[nelm6][4];
         nei6=new int[nelm6][kcm];
         jnb6=new int[nelm6];
           xn6=new double[nelm6];
           yn6=new double[nelm6];
           zn6=new double[nelm6];
        area6=new double[nelm6];

        for(i=1;i<=node6;i++){for(j=1;j<=kcm;j++){nei6[i][j]=0;}}

        for(i=1;i<=nelm6;i++){for(j=1;j<=4;j++){mtj6[i][j]=mtj1[i][j];}}
        for(i=1;i<=nelm6;i++){for(j=1;j<=4;j++){jac6[i][j]=jac1[i][j];}}


        for(i=1;i<=node6;i++) {px6[i]=px1[i];py6[i]=py1[i];}

        for(i=1;i<=node6;i++) {
                AnyDepth((int)px6[i],(int)py6[i],z_xy) ;
                pz6[i]=z_xy;}

        normal(node6,nelm6,px6,py6,pz6,mtj6,xn6,yn6,zn6,area6);
```

```
        for(i=1;i<=node1;i++) {pz1[i]=0;}

        normal(node1,nelm1,px1,py1,pz1,mtj1,xn1,yn1,zn1,area1);

}// seabed

//////////////////////////////////////////水平面元素自動生成
////////////////////////////////////////////////
public void squareMesh(){

trmodel(nin,ibex,ibin,ibno1,nob,nib,index,node1,jnb1,nei1,px1,py1,nelm1,mtj1,jac1,i
dm,ifix,delx);
            quexelm(node1,nelm1,mtj1,jac1,kv,idm,map,px1,py1);
            quisogen(node1,nelm1,mtj1,jac1,px1,py1,id,mmtj,ifix,idm);
            changemtj(nelm1,mtj1,px1,py1); //對各元素配合 Gauss 積分轉換成第 1
節點在左下角
            Seabed() ;
} //squaremesh

public void triangleWriteData()    {

     // AxisChange();

       try{ pw_tridata=new PrintWriter(new BufferedWriter(new
FileWriter(file_tridata)));
            pw_tridata.printf("%10d\n",node);
            pw_tridata.printf("%10d\n",nin);
            pw_tridata.printf("%10d\n",nelm1);
            pw_tridata.printf("%10d\n",ibex);
            pw_tridata.printf("%10d\n",nob);
            pw_tridata.printf("%10d\n",pseudoN1);
            pw_tridata.printf("%10d\n",pseudoN2);
            pw_tridata.printf("%10d\n",pseudoN3);
            pw_tridata.printf("%10d\n",pseudoN);
            pw_tridata.printf("%10d\n",nib);
            pw_tridata.printf("%10d\n",totalN1);
            pw_tridata.printf("%10d\n",totalInn);
```

```
for(i=1;i<=nelm1;i++){pw_tridata.printf("%10d%10d%10d\n",mtj1[i][1],mtj1[i][2],mtj
1[i][3]);}

for(i=1;i<=nelm1;i++){pw_tridata.printf("%10d%10d%10d\n",jac1[i][1],jac1[i][2],jac1[
i][3]);}
        for(i=1;i<=nelm1;i++){pw_tridata.printf("%10\n",idm[i]);}

for(i=1;i<=node;i++){pw_tridata.printf("%10d%10d%10.3f%10.3%10.3\n",ibno1[i],ifix
[i],px1[i],py1[i],pz1[i]);}

        if(nin>0){
                for(i=1;i<=nin;i++) {pw_tridata.printf("%10d\n",ibin[i]); }
        }
    pw_tridata.close();}
    catch(Exception e) {tf_command1.setText("WriteData");}
}
public void trmodel(int nin,int ibex,int []ibin,int []ibno,int nob,int nib,int []index,int
node,int []jnb,int [][]nei,
                        double []px,double []py,int nelm,int [][]mtj,int [][]jac,int
[]idm,int []ifix,double delx){

        for(i=1;i<=kbd+1;i++){index[i]=0; }     //初期化

        node=0;
        nelm=0;
        for(i=1;i<=kte;i++){
            idm[i]=0;
            for(j=1;j<=3;j++) {mtj[i][j]= 0; jac[i][j]= 0;}}

        for(i=1;i<=ktj;i++){
            jnb[i]= 0;
            for(j=1;j<=kcm;j++){ nei[i][j]= 0;}}

        for(i=1;i<=ktj;i++){ifix[i]= 0;}

        ntp = nob + nib;
        xmin = px[1];
```

```
xmax = px[1];
ymin = py[1];
ymax = py[1];
for(i=2;i<=ntp;i++){
      if(px[i] < xmin) xmin = px[i];
      if(xmax < px[i]) xmax = px[i];
      if(py[i] < ymin) ymin = py[i];
      if(ymax < py[i]) ymax = py[i];}

rax = xmax - xmin;
ray = ymax - ymin;
dmax = rax;
if(dmax < ray) dmax = ray;

for(i=1;i<=ntp;i++){
      px[i] = (px[i] - xmin) / dmax;
      py[i] = (py[i] - ymin) / dmax;}    //正規化

nelm = 1;
ia = ktj + 1;
ib = ktj + 2;
ic = ktj + 3;
mtj[1][1]= ia;          mtj[1][2]= ib;          mtj[1][3]= ic;
jac[1][1]= 0;           jac[1][2]= 0;           jac[1][3]= 0;

px[ia]=-1.23;          //起始三角形位置(任意値)
py[ia]=-0.5;
px[ib]= 2.23;
py[ib]=-0.5;
px[ic]= 0.5;
py[ic]= 2.5;


rough(nin,ibex,ibin,ibno,ntp,nib,node,px,py,jnb,nei,nelm,mtj,jac,idm,list,iadres,istack,
kv,ibr,map);

      for(i=1;i<=node;i++){ifix[i]=1;}
```

```
            trfine(node,px,py,jnb,nei,nelm,mtj,jac,idm,iadres,istack);

           //
trfine(xmin,ymin,dmax,node,px,py,jnb,nei,nelm,mtj,jac,idm,iadres,istack,delx);

            if(nelm!=(2*node+1)) {tf_command.setText("發生錯誤中止執行
trmodel"); }

            remove(index,nelm,mtj,jac,idm);
            check(nelm,mtj,jac);

            for(i=1;i<=node;i++){
                  px[i]= px[i]* dmax + xmin;
                  py[i]= py[i]* dmax + ymin;}
} //trmodel

public void rough(int nin,int ibex,int []ibin,int []ibno,int ntp,int nib,int node,double
[]px,double []py,
                        int []jnb,int [][]nei,int nelm,int [][]mtj,int [][]jac,int []idm,
                        int []list,int []iadres,int []istack,int []kv,int []ibr,int []map){

                  jz=0;
                  nb=1;
                  np = ibex;
                  for(j=1;j<=np;j++) {list[j]= ibno[j]; }


bougen(jz,np,list,ntp,px,py,jnb,nei,nelm,mtj,jac,idm,iadres,istack,kv,ibr,map,node);

                  for(k= 1;k<=nelm;k++){
                        if(idm[k]==jz){
                              ma=iadres[mtj[k][1]];
                              mb=iadres[mtj[k][2]];
                              mc=iadres[mtj[k][3]];
                              ms=ma*mb*mc;
                              mp=(mb-ma)*(mc-mb)*(ma-mc);
                              if(ms!=0&&0< mp) idm[k]=nb; }
                     }
```

```
            for(i=1;i<=nin;i++){
                  nb = i;
                  np = ibin[nb];
                  for(j=1;j<=np;j++){list[j]= node + j;    }
                  js = list[1];
                  xs = px[js];
                  ys = py[js];
                  locate(xs,ys,px,py,mtj,jac,nelm,loc);
                  jz = idm[loc];

bougen(jz,np,list,ntp,px,py,jnb,nei,nelm,mtj,jac,idm,iadres,istack,kv,ibr,map,node);

                  for(k= 1;k<=nelm;k++){
                        if(idm[k]==jz){
                              ma = iadres[mtj[k][1]];
                              mb = iadres[mtj[k][2]];
                              mc = iadres[mtj[k][3]];
                              ms = ma * mb * mc;
                              mp = (mb - ma) * (mc - mb) * (ma - mc);
                              if((ms != 0)&& (mp < 0))    idm[k]= 0;}
               }}

            for(i=1;i<=ktj+3;i++){iadres[i]= 0; }

            for(i=1;i<=nib;i++){
                  node = node + 1;
                  xa = px[node];
                  ya = py[node];
                  locate(xa,ya,px,py,mtj,jac,nelm,it);
                  jz = idm[it];

delaun(jz,node,node,ntp,px,py,jnb,nei,nelm,mtj,jac,idm,iadres,istack); }

            if(node!=ntp) {tf_command.setText("發生錯誤中止執行 rough");     }
 }   //rough

public void bougen(int jz,int np,int []list,int ntp,double []px,double []py,int []jnb,int
```

```
[][]nei,
                        int nelm,int [][]mtj,int [][]jac,int []idm,int []iadres,int
[]istack,int []kv,int []ibr,
                        int []map,int node){
        inp = 0;
        for(i=1;i<=ktj+3;i++){iadres[i]= 0; }
        for(i=1;i<=np      ;i++){iadres[list[i]]=i; }

        js = list[1];
        if(js>node)
{inp=inp+1;delaun(jz,js,js,ntp,px,py,jnb,nei,nelm,mtj,jac,idm,iadres,istack); }

bougen30:for(i=1;i<=np;i++){
                ip = list[Mod(i,np)+1];
                iq = list[i];
                if(ip>node&&i!=np)
{inp=inp+1;delaun(jz,ip,ip,ntp,px,py,jnb,nei,nelm,mtj,jac,idm,iadres,istack); }
                for(j=1;j<=jnb[iq];j++){
                for(k=1;k<=3;k++){if(mtj[nei[iq][j]][k]==ip)    continue bougen30;}}
                search(jz,ip,iq,jnb,nei,nelm,mtj,jac,idm,istack,iv,kv,iadres,ibr);
                poly(iq,ip,iv,kv,px,py,nelm,mtj,jac,jnb,nei,map);
        }
        node = node + inp;
}//bougen

  public void delaun(int jz,int js,int jg,int ntp,double []px,double []py,int []jnb,int
[][]nei,int nelm,int [][]mtj,
                        int [][]jac,int []idm,int []iadres,int []istack) {

        itop=0;
        maxstk=ntp;
        for(i=js;i<=jg;i++){
            ip=i;
            xp=px[ip];
            yp=py[ip];

            locate(xp,yp,px,py,mtj,jac,nelm,itri);
```

```
if(idm[itri]!=jz) {tf_command.setText("發生錯誤中止執行 delaun"); }

ia =jac[itri][1];
ib =jac[itri][2];
ic =jac[itri][3];
iv1=mtj[itri][1];
iv2=mtj[itri][2];
iv3=mtj[itri][3];
mtj[itri][1]=ip;
mtj[itri][2]=iv1;
mtj[itri][3]=iv2;
jac[itri][1]=nelm+2;
jac[itri][2]=ia;
jac[itri][3]=nelm+1;

nelm = nelm + 1;
idm[nelm]= jz;
mtj[nelm][1]= ip;
mtj[nelm][2]= iv2;
mtj[nelm][3]= iv3;
jac[nelm][1]= itri;
jac[nelm][2]= ib;
jac[nelm][3]= nelm + 1;
nelm = nelm + 1;
idm[nelm]= jz;
mtj[nelm][1]= ip;
mtj[nelm][2]= iv3;
mtj[nelm][3]= iv1;
jac[nelm][1]= nelm - 1;
jac[nelm][2]= ic;
jac[nelm][3]= itri;

incr(iv1,nelm    ,jnb,nei);
incr(iv2,nelm-1,jnb,nei);
if(iv3 <= ktj)    nei[iv3][Neibor(iv3,itri,jnb,nei)]=nelm-1;
incr(iv3,nelm,jnb,nei);
jnb[ip] = 3;
nei[ip][1] = itri;
```

```
nei[ip][2] = nelm - 1;
nei[ip][3] = nelm;
if(ia!=0){
        ms = iadres[iv1]*iadres[iv2];
        idf = Math.abs(iadres[iv1]-iadres[iv2]);
        if(idm[ia]==jz&& (ms==0||idf!= 1)) {
                itop = itop + 1;
                istack[itop]= Ipush(itri,maxstk,itop); }
}
if(ib!=0){
        edge(ib, itri, jac, iedge);
        jac[ib][iedge]= nelm - 1;
        ms = iadres[iv2]*iadres[iv3];
        idf = Math.abs(iadres[iv2]-iadres[iv3]);
        if(idm[ib]==jz&&(ms== 0||idf!= 1)){
                itop = itop + 1;
                istack[itop] = Ipush(nelm-1, maxstk, itop); }
  }
if(ic!=0){
        edge(ic,itri,jac,iedge);
        jac[ic][iedge]= nelm;
        ms = iadres[iv3]* iadres[iv1];
        idf = Math.abs(iadres[iv3]- iadres[iv1]);
        if(idm[ic]==jz&&(ms==0||idf!= 1)){
                itop = itop + 1;
                istack[itop]= Ipush(nelm,maxstk,itop); }
}

while(itop>0){
        il=istack[itop];
        itop=itop-1;
        ir= jac[il][2];
        edge(ir,il,jac,ierl);
        iera=Mod(ierl,3)+1;
        ierb=Mod(iera,3)+1;
        iv1=mtj[ir][ierl];
        iv2=mtj[ir][iera];
        iv3=mtj[ir][ierb];
```

```
swap(px[iv1],py[iv1],px[iv2],py[iv2],px[iv3],py[iv3],xp,yp,iswap);
if(iswap==1){
        ia = jac[ir][iera];
        ib = jac[ir][ierb];
        ic = jac[il][3];
        mtj[il][3]= iv3;
        jac[il][2]= ia;
        jac[il][3]= ir;
        mtj[ir][1]= ip;
        mtj[ir][2]= iv3;
        mtj[ir][3]= iv1;
        jac[ir][1]= il;
        jac[ir][2]= ib;
        jac[ir][3]= ic;
        decr(iv1,il,jnb,nei);
        decr(iv2,ir,jnb,nei);
        incr(ip,ir,jnb,nei);
        incr(iv3,il,jnb,nei);

        if(ia!=0){
                edge(ia,ir,jac,iedge);
                jac[ia][iedge] = il;
                ms = iadres[iv2]*iadres[iv3];
                idf = Math.abs(iadres[iv2]-iadres[iv3]);
                if((idm[ia]== jz) && ((ms==0) || (idf!=1))){
                        itop = itop + 1;
                        istack[itop] = Ipush(il, maxstk, itop); }
        }
        if(ib!=0){
                ms = iadres[iv3]*iadres[iv1];
                idf = Math.abs(iadres[iv3]-iadres[iv1]);
                if((idm[ib]==jz) && ((ms==0) || (idf!=1))){
                        itop = itop + 1;
                        istack[itop]= Ipush(ir, maxstk, itop); }
        }
        if(ic!=0){
                edge(ic,il,jac,iedge);
                jac[ic][iedge] = ir;}
```

```
                        }
                    }
            }    //i
  } //delaun

public void locate(double xp,double yp,double []px,double []py,int [][]mtj,int
[][]jac,int nelm,int itri) {
            itri = nelm;
locate10:
            for(i=1;i<=3;i++){
                ia=mtj[itri][i];
                ib=mtj[itri][Mod(i,3)+1];
                if((px[ia]-xp)*(py[ib]-yp)<(py[ia]-yp)*(px[ib]-xp)) {itri=jac[itri][i];
continue locate10;}
            }
} //locate

public void search(int iz,int ip,int iq,int []jnb,int [][]nei,int nelm,int [][]mtj,int [][]jac,
                                        int []idm,int []istack,int iv,int []kv,int []iadres,int
[]ibr)    {

            iv    = 0;
            mstk= 0;
            nbr = 0;
            for(i=1;i<=nelm;i++){kv[i]=0;ibr[i]=0;}
            for(i=1;i<=ktj;i++){istack[i]= 0; }
            for(i=1;i<=jnb[iq];i++){nbr=nbr+1; ibr[nei[iq][i]]=nbr;}

            for(i=1;i<=jnb[iq];i++){
                ielm = nei[iq][i];
                j = Ivert(ielm,iq,mtj);
                ja=Mod(j,3)+ 1;
                jb=Mod(ja,3)+ 1;
                ia= mtj[ielm][ja];
                ib= mtj[ielm][jb];
                idf=Math.abs(iadres[ia]-iadres[ib]);
                ms = iadres[ia]*iadres[ib];
                if((idf == 1) && (ms != 0)) continue;
```

```
            jelm = jac[ielm][ja];
            if(jelm==0||idm[jelm]!=iz)    continue;
            nbr=nbr+1;
            ibr[jelm]=nbr;
            if(mtj[jelm][1]==ip||mtj[jelm][2]==ip||mtj[jelm][3]==ip) break;
            mstk = mstk + 1;
            istack[mstk]= jelm;}

while(true){
        iv=iv+1;
        kv[iv]= jelm;
        if(ibr[jelm]<=jnb[iq]) break;
        min = kte + 1;
            for(j=1;j<=3;j++){
                jr = jac[jelm][j];
                if(jr==0 ||ibr[jr]==0) continue;
                ja = Mod(j,3)+1;
                ia = mtj[jelm][j];
                ib = mtj[jelm][ja];
                idf= Math.abs(iadres[ia] - iadres[ib]);
                ms = iadres[ia]*iadres[ib];
                if(idf==1&&ms!=0) continue;
                if(ibr[jr]<min) {kelm=jr;    min=ibr[jr];}
        }
        jelm = kelm;
 } //true
 while(true){
        if(mstk==0){tf_command.setText("發生錯誤中止執行 search1");    }
        ielm = istack[1];
        mstk = mstk - 1;
        for(i=1;i<=mstk;i++){istack[i]=istack[i+1];}
        istack[mstk+1]=0;
        for(j=1;j<=3;j++){
            ja =Mod(j,3)+ 1;
            ia= mtj[ielm][j];
            ib= mtj[ielm][ja];
            idf=Math.abs(iadres[ia]-iadres[ib]);
            ms = iadres[ia]*iadres[ib];
```

```
                if(idf==1&&ms!=0) continue;
                jelm = jac[ielm][j];
                if(jelm==0||ibr[jelm]!=0||idm[jelm]!=iz) continue;
                nbr = nbr + 1;
                ibr[jelm]= nbr;
                if(mtj[jelm][1]==ip||mtj[jelm][2]==ip||mtj[jelm][3]==ip) break;
                mstk = mstk + 1;
                istack[mstk]= jelm;}
        }   //true

}   //search

public void    poly(int iq,int ip,int iv,int []kv,double []px,double []py,int nelm,int
[][]mtj,
                        int [][]jac,int []jnb,int [][]nei,int []map)    {

            if(iv == 2){
                  edge(kv[1], kv[2], jac, ia);
                  edge(kv[2], kv[1], jac, ja);
                ir = jac[kv[1]][Mod(ia,3)+1];
                jr = jac[kv[2]][Mod(ja,3)+1];
                mtj[kv[1]][Mod(ia,3)+1] = iq;
                jac[kv[1]][ia]= jr;
                jac[kv[1]][Mod(i,np)+1] = kv[2];
                mtj[kv[2]][Mod(i,np)+1] = ip;
                jac[kv[2]][ja] = ir;
                jac[kv[2]][Mod(i,np)+1] = kv[1];

                if(ir != 0 ){ edge(ir, kv[1], jac, iedge); jac[ir][iedge] = kv[2];}
                if(jr != 0 ){ edge(jr, kv[2], jac, iedge); jac[jr][iedge] = kv[1];}

                iv1 = mtj[kv[1]][ia];
                iv2 = mtj[kv[2]][ja];
                  decr(iv1, kv[2], jnb, nei);
                  decr(iv2, kv[1], jnb, nei);
                  incr(iq, kv[1], jnb, nei);
                  incr(ip, kv[2], jnb, nei);
            }
```

```
if(iv != 2){
        npa = 0;
        npb = 0;
        for(i=1;i<=lte+2;i++){nsra[i]= 0; nsrb[i]= 0;}
        for(i=1;i<=nelm; i++){map[i]        = 0;}
        for(i=1;i<=iv;      i++){map[kv[1]]= 1;}

        for(i=1;i<=iv;i++){
                ielm = kv[i];
                for(j=1;j<=3;j++){ivx = mtj[ielm][j];    decr(ivx, ielm, jnb, nei);}
        }

          pick(iq, ip, iv, kv, mtj, jac, map, npa, npb, nsra, nsrb);
          subdiv(npa, nsra, px, py, nta, iena, jeea, ihen, jhen, iad, jstack);
          subdiv(npb, nsrb, px, py, ntb, ienb, jeeb, ihen, jhen, iad, jstack);

        if(iv!=nta+ntb ){tf_command.setText("發生錯誤中止執行 poly"); }

        for(i=1;i<=iv;i++){
                for(j=1;j<=3;j++){
                        jac[kv[i]][j]= 0;}}
        for(i=1;i<=nta;i++){
                ielm = kv[i];
        for(j=1;j<=3;j++){
                        mtj[ielm][j]= iena[i][j];
                        if(jeea[i][j]!=0 )    jac[ielm][j] = kv[jeea[i][j]];}}
        for(i=1;i<=ntb;i++){
                ielm = kv[nta + i];
        for(j=1;j<=3;j++){
                        mtj[ielm][j] = ienb[i][j];
                        if(jeeb[i][j]!=0) jac[ielm][j]= kv[nta + jeeb[i][j]]; }}
        for(i=1;i<=iv;i++){
                ielm = kv[i];
        for(j=1;j<=3;j++){
                        ivx = mtj[ielm][j];
                          incr(ivx, ielm, jnb, nei);}}
        for(i=1;i<=iv;i++){
```

```
                    ielm = kv[i];
    outside: for(j=1;j<=3;j++){
                        jelm = jac[ielm][j];
                        if(jelm!= 0) continue;
                        ips = mtj[ielm][j];
                        ipg = mtj[ielm][Mod(j,3)+1];
                        for(k=1;k<=jnb[ipg];k++){
                            kelm = nei[ipg][k];
                        for(l=1;l<=3;l++){
                            iva = mtj[kelm][l];
                            ivb = mtj[kelm][Mod(l,3)+ 1];
                            if((iva==ipg) && (ivb==ips))
{jac[ielm][j]=kelm;jac[kelm][l]=ielm;continue outside;}
                          }}     //k,l
                    }}   //i,j
            }
}   //poly

public void    pick(int iq,int ip,int iv,int []kv,int [][]mtj,int [][]jac,int []map,int npa,int
npb,int []nsra,int []nsrb) {

            npa = 1;
            nsra[npa] = ip;
            ivx = Ivert(kv[1],ip,mtj);
            npa = npa + 1;
            nsra[npa] = mtj[kv[1]][Mod(ivx,3)+1];
            for(i=2;i<=iv-1;i++){
                jvx = Ivert(kv[i], nsra[npa], mtj);
                jelm = jac[kv[i]][jvx];
                if(jelm==       0){npa=npa+1; nsra[npa]=mtj[kv[i]][Mod(jvx,3)+1];}
                if(map[jelm]==0){npa=npa+1; nsra[npa]=mtj[kv[i]][Mod(jvx,3)+1];}
             }
            npa = npa + 1;
            nsra[npa]= iq;
            npb = 1;
            nsrb[npb]= iq;
            ivx = Ivert(kv[iv], iq, mtj);
            npb = npb + 1;
```

```
            nsrb[npb]= mtj[kv[iv]][Mod(ivx,3)+1];
            for(i=iv-1;i<=2;i--){
                  jvx = Ivert(kv[i], nsrb[npb], mtj);
                  jelm = jac[kv[i]][jvx];
                  if(jelm==0)       {npb=npb+1; nsrb[npb]=mtj[kv[i]][Mod(jvx,3)+1];}
                  if(map[jelm]==0){npb=npb+1; nsrb[npb]=mtj[kv[i]][Mod(jvx,3)+1];}
            }
            npb = npb + 1;
            nsrb[npb]= ip;
            if(iv!=npa+ npb-4){tf_command.setText("發生錯誤中止執行 pick"); }
}   //pick

public void    subdiv(int npl,int []nsr,double []px,double []py,int nte,int [][]ien,int
[][]jee,int [][]ihen,int []jhen,
                                    int []iad,int []jstack) {

            nte = 0;
            nnpl= npl;
            for(i=1;i<=lte;i++){for(j=1;j<=3;j++){ien[i][j]=0;jee[i][j]=0;}}

            while(3<=nnpl){
                  nbs = 1;
                  while(nbs<=nnpl-1){
                        ia = nsr[nbs];
                        ib = nsr[nbs + 1];
                        ic = nsr[Mod((nbs+1),nnpl)+1];
                        xa = px[ib]- px[ia];
                        ya = py[ib]- py[ia];
                        xb = px[ic]- px[ia];
                        yb = py[ic]- py[ia];
                        see = xa * yb - xb * ya;
                        if(see>0.000000000001){
                              nte = nte + 1;
                              ien[nte][1]= ia;
                              ien[nte][2]= ib;
                              ien[nte][3]= ic;
                              nnpl = nnpl - 1;
                              for(i=nbs+1;i<=nnpl;i++){nsr[i]=nsr[i+1]; }
```

```
                    }
                nbs = nbs + 1;}
            }

            ix = 0;
            for(i=1;i<=2*lte+1;i++){
                ihen[i][1]= 0;
                ihen[i][2]= 0;
                jhen[i]= 0;
                iad[i] = 0;}

            for(i=1;i<=nte;i++){
 subdiv70: for(j=1;j<=3;j++){
                    ia = ien[i][j];
                    ib = ien[i][Mod(j,3)+1];
                    for(k=1;k<=ix;k++){
                        if((ihen[k][1]==ib) && (ihen[k][2]==ia)){
                            jee[i][j]= jhen[k];
                            jee[jhen[k]][iad[k]] = i;
                            continue subdiv70; }
                    }
                    ix = ix + 1;
                    jhen[ix]= i;
                    iad[ix] = j;
                    ihen[ix][1]= ia;
                    ihen[ix][2]= ib;
                }}    //i,j

              lawson(nte,ien,jee,npl,px,py,jstack);

}    //subdiv

public void    lawson(int nte,int [][]ien,int [][]jee,int npl,double []px,double []py,int
[]jstack) {

            itop = 0;
            maxstk = npl;
            ncount = 0;
```

```
        for(i=1;i<=nte;i++){
            ielm = i;
            itop = itop + 1;
            jstack[itop]= Ipush(ielm, maxstk, itop); }

lawson10: while(0<itop) {
            ncount = ncount + 1;
            if(lte < ncount){
                tf_command.setText("發生錯誤中止執行 lawson");
                }
            il = jstack[itop];
            itop = itop - 1;
            for(j=1;j<=3;j++){
                jl1 = j;
                jl2 =Mod(nnpl,3)+1;
                jl3 =Mod(jl2,3)+1;
                ir = jee[il][jl1];
                if(ir==0) continue;
                iv1 = ien[il][jl1];
                iv2 = ien[il][jl2];
                iv3 = ien[il][jl3];
                xxl = px[ien[il][jl3]];
                yyl = py[ien[il][jl3]];
                edge(ir,il,jee,jr1);
                jr2 =Mod(jr1,3)+ 1;
                jr3 =Mod(jr2,3)+ 1;
                iv4 = ien[ir][jr3];
                  swap(px[iv2],py[iv2],px[iv1],py[iv1],px[iv4],py[iv4],xxl,yyl,iswap);
                if(iswap == 1){
                    ia = jee[il][jl2];
                    ib = jee[ir][jr2];
                    ien[il][jl2]= iv4;
                    jee[il][jl1]= ib;
                    jee[il][jl2]= ir;
                    ien[ir][jr2]= iv3;
                    jee[ir][jr1]= ia;
                    jee[ir][jr2]= il;
```

```
                        if(ia!=0){ edge(ia,il,jee,iedge); jee[ia][iedge]= ir;}
                        if(ib!=0){ edge(ib,ir,jee,iedge); jee[ib][iedge]= il;}
                        itop = itop + 1;
                        jstack[itop]= Ipush(il,maxstk,itop);
                        continue lawson10; }
                  }   //j
               } //lawson10
}   //lawson

public void trfine_old(double xmin,double ymin,double dmax,int node,double
[]px,double []py,int []jnb,
                        int [][]nei,int nelm,int [][]mtj,int [][]jac,int []idm,int
[]iadres,int []istack,double delx){

         range = dmax;
         pa1 = 0.0;
         pb1 = 0.0;
         pa2 = 1.0;
         pb2 = 0.0;
         pa3 = 1.0;
         pb3 = 1.0;
         pa4 = 0.0;
         pb4 = 1.0;
         jz = 1;
         js = node + 1;
         if(dmax>=delx){
                  ndiv =(int)(range/delx);
                  delta=1.0/(double)ndiv;
                  dltd = delta / alpha;
                  dltd = dltd * dltd;
                  for(k=1;k<=ndiv+1;k++){
                        xg = pa2 + (pa3 - pa2) / (double)ndiv*(k - 1);
                        xs = pa1 + (pa4 - pa1) / (double)ndiv*(k - 1);
                        yg = pb2 + (pb3 - pb2) / (double)ndiv*(k - 1);
                        ys = pb1 + (pb4 - pb1) / (double)ndiv*(k - 1);
                        for(j=1;j<=ndiv+1;j++){
                              xp = xs + (xg - xs) / (double)ndiv*(j - 1);
                              yp = ys + (yg - ys) / (double)ndiv*(j - 1);
```

```
                        if(Trplace(jz,xp,yp,nelm,mtj,jac,idm,px,py,dltd)){
                            node = node + 1;
                             px[node] = xp;
                             py[node] = yp;
                             AnyDepth((int)xp,(int)yp,z_xy);
                             pz1[node]=xy_z;}
                    }
                }

delaun(jz,js,node,node,px,py,jnb,nei,nelm,mtj,jac,idm,iadres,istack);
            }
} //trfine

public void trfine(int node,double []px,double []py,int []jnb,
                    int [][]nei,int nelm,int [][]mtj,int [][]jac,int []idm,int
[]iadres,int []istack){
        if(innerNode!=0) js=innNodeStart;
        if(innerNode!=0&&innerPoint==0) jg=innNodeEnd;
        if(innerNode==0&&innerPoint!=0) js=innPointStart;
        if(innerPoint!=0) jg=innPointEnd;

        if(innerNode!=0||innerPoint!=0)
delaun(1,js,jg,node,px,py,jnb,nei,nelm,mtj,jac,idm,iadres,istack);

} //trfine


public boolean Trplace(int iz,double xp,double yp,int nelm, int [][]mtj,int [][]jac,int
[]idm,
                        double []px,double []py,double dltd){    //check

        boolean trplace=true;
        locate(xp,yp,px,py,mtj,jac,nelm,loc);

        if(iz==idm[loc]){
            for(j=1;j<=3;j++){
                ia = mtj[loc][j];
                xa = px[ia];
```

```
                    ya = py[ia];
                    dpp = (xa-xp)*(xa-xp)+(ya-yp)*(ya-yp);
                    if(dpp < dltd) trplace = false;
                    if(dpp >=dltd){
                    jr = jac[loc][j];
                    if(iz!=idm[jr]){
                        ib = mtj[loc][Mod(j,3)+1];
                        xb = px[ib];
                        yb = py[ib];
                        eps = (yb-ya)*xp-(xb-xa)*yp-xa*(yb-ya)+ya*(xb-xa);
                        eps = eps * eps;
                        eps0= (yb-ya)*(yb-ya)+(xb-xa)*(xb-xa);
                        epl = eps / eps0;
                        if(epl< dltd)trplace = false;
                        if(epl>=dltd)trplace = true;
                     }
                    }
                  } //j
            }
                if(iz!=idm[loc]) trplace = false;
          return trplace;

} //Trplace


public void    remove(int []index,int nelm,int [][]mtj,int [][]jac,int []idm)    {

            ielm = 0;
            index[1]= 1;
            for(i = 1;i<=nex;i++){
                iz = i;
                inelm = 0;
                for(j = index[i];j<=nelm;j++){
                    if(idm[j]== iz){
                        ielm = ielm + 1;
                        jelm = j;
                        inelm = inelm + 1;
                        if(ielm!=jelm){
```

```
                                    for(k = 1;k<=3;k++){
                                        mkp[k]= mtj[ielm][k];
                                        jkp[k]= jac[ielm][k]; }

                                    ikp = idm[ielm];
                                    for(k = 1;k<=3;k++){
                                        kelm = jac[jelm][k];

if(kelm!=0){ edge(kelm,jelm,jac,kedg);jac[kelm][kedg]=ielm+kte+1;}
                                    }
                                    for(l=1;l<=3;l++){
                                        lelm = jkp[l];

if(lelm!=0){ edge(lelm,ielm,jac,ledg);jac[lelm][ledg]=jelm+kte+1;}
                                    }
                                    for(k = 1;k<=3;k++){
                                        jac[ielm][k]=Mod(jac[ielm][k],(kte+1));
                                        jac[jelm][k]=Mod(jac[jelm][k],(kte+1));
                                        kelm = jac[ielm][k];
                                        lelm = jac[jelm][k];
                                        for(l= 1;l<=3;l++){
                                            jac[kelm][l]= Mod(jac[kelm][l],(kte+1));
                                            jac[lelm][l]= Mod(jac[lelm][l],(kte+1));}
                                    }
                                    for(k = 1;k<=3;k++){jkp[k]= jac[ielm][k]; }
                                    for(k = 1;k<=3;k++){
                                        mtj[ielm][k]= mtj[jelm][k];
                                        jac[ielm][k]= jac[jelm][k];
                                        mtj[jelm][k]= mkp[k];
                                        jac[jelm][k]= jkp[k];}
                                    idm[ielm]= idm[jelm];
                                    idm[jelm]= ikp;
                                }}
                            }   //j
                        index[i+1]= index[i]+ inelm;
                    }   //i

                for(i = 1;i<=ielm;i++){
```

```
            for(j = 1;j<=3;j++){if(ielm<jac[i][j]) jac[i][j]= 0;}}
        for(i = ielm + 1;i<=nelm;i++){
            for(j=1;j<=3;j++){mtj[i][j]=0;jac[i][j]=0;}}

        nelm = ielm;
}   //remove

public void check(int nelm,int [][]mtj,int [][]jac) {

        for(i=1;i<=nelm;i++){
            for(j=1;j<=3;j++){
                ielm=i;
                ia=mtj[i][j];
                ib=mtj[i][Mod(j,3)+1];
                jelm=jac[i][j];
                if(jelm==0) continue ;
                for(k=1;k<=3;k++){
                    kelm=jac[jelm][k];
                    if(kelm==ielm) {
                        ja=mtj[jelm][k];
                        jb=mtj[jelm][Mod(k,3)+1];
                        if((ia==jb) && (ib==ja) ) break;
                   tf_command.setText("發生錯誤中止執行 check1"); }
                }
                   tf_command.setText("發生錯誤中止執行 check2");
        }}
}   //check

public void incr(int n,int l,int []jnb,int [][]nei)    {
        if(n <= ktj){
            jnb[n]= jnb[n]+1;
            if(jnb[n]>kcm){ tf_command.setText("發生錯誤中止執行 incr"); }
            nei[n][jnb[n]] = l; }
}   //incr

public void decr(int n,int l,int []jnb,int [][]nei)    {

        if(n <= ktj){
```

```
                inb = Neibor(n, l, jnb, nei);
                for(i=inb;i<=jnb[n]-1;i++){
                    nei[n][i]= nei[n][i+1];
                    nei[n][jnb[n]]=0;
                    jnb[n]=jnb[n]-1;
                if(jnb[n]==0){tf_command.setText("發生錯誤中止執行 decr"); }
            }
        }
}   //decr

public void    swap(double x1,double y1,double x2,double y2,double x3,double
y3,double xp,double yp,int iswap) {
            x13 = x1 - x3;
            y13 = y1 - y3;
            x23 = x2 - x3;
            y23 = y2 - y3;
            x1p = x1 - xp;
            y1p = y1 - yp;
            x2p = x2 - xp;
            y2p = y2 - yp;

            cosa = x13 * x23 + y13 * y23;
            cosb = x2p * x1p + y1p * y2p;
            if(cosa>=0 && cosb>=0)    iswap=0;
            if(cosa< 0 && cosb< 0)    iswap=1;

            sina = x13*y23-x23*y13;
            sinb = x2p*y1p-x1p*y2p;
            if((sina*cosb+sinb*cosa)< 0) iswap=1;
            if((sina*cosb+sinb*cosa)>=0) iswap=0;

} //swap

public void    edge(int l,int k,int [][]jac,int iedge) {
            iedge = -1;
            for(i=1;i<=3;i++){ if(jac[l][i]== k){iedge=i; break; }    }
            if(iedge== -1){ tf_command.setText("發生錯誤中止執行 edge");    }
}//edge
```

```
public int Ipush(int item,int maxstk,int itop) {
        if(maxstk < itop){tf_command.setText("發生錯誤中止執行  Ipush");  }
        if(maxstk >=itop){ii=item;}
        return ii;
  }    //Ipush

public int Neibor(int n,int l,int []jnb,int [][]nei) {
      int neibor;
        neibor = -1;
        for(i=1;i<=jnb[n];i++) {if(nei[n][i]==l) {ii=i;break; }          }
        if(neibor==-1) {tf_command.setText("發生錯誤中止執行  Neibor");  }
        return ii;
  } //Neibor

public int Ivert(int l,int k,int [][]mtj) {
      int ivert;
        ivert = -1;
        for(i=1;i<=3;i++){if(mtj[l][i]==k){ii=i;break; }    }
        if(ivert == -1) {tf_command.setText("發生錯誤中止執行  Ivert");}
        return ii;
} //Ivert


public void quexelm(int node,int nelm,int [][]mtj,int [][]jac,int []kv,int []idm,int
[]map,double []px,double []py){

        iv = 0;
        for(i=1;i<=nelm;i++){
            if(mtj[i][1]==0) continue;
            if(mtj[i][4]!=0) continue;
            pxa = px[mtj[i][1]] - px[mtj[i][2]];
            pya = py[mtj[i][1]] - py[mtj[i][2]];
            pxb = px[mtj[i][2]] - px[mtj[i][3]];
            pyb = py[mtj[i][2]] - py[mtj[i][3]];
            pxc = px[mtj[i][3]] - px[mtj[i][1]];
            pyc = py[mtj[i][3]] - py[mtj[i][1]];
            pl[1] = pxa * pxa + pya * pya;
```

```
                pl[2] = pxb * pxb + pyb * pyb;
                pl[3] = pxc * pxc + pyc * pyc;

                for(j=1;j<=3;j++) {if (pl[Mod(j,3)+1]<pl[j]&&pl[Mod(j+1,3)+1]<
pl[j]){ ip=j;    break;} }

                jn = jac[i][ip];
                if(idm[i]!=idm[jn]) continue;
                if(mtj[jn][4]!=0)      continue;
                if(jn==0)                    continue;
                quedge(jn,i, jac, ie1);
                ie2 =Mod(ie1,3) + 1;
                ie3 =Mod(ie2,3) + 1;
                px12 = px[mtj[jn][ie2]] - px[mtj[jn][ie3]];
                py12 = py[mtj[jn][ie2]] - py[mtj[jn][ie3]];
                px23 = px[mtj[jn][ie3]] - px[mtj[jn][ie1]];
                py23 = py[mtj[jn][ie3]] - py[mtj[jn][ie1]];
                pl12 = px12 * px12 + py12 * py12;
                pl23 = px23 * px23 + py23 * py23;
                if(pl[ip]< pl12||pl[ip]< pl23) continue;

                ia = mtj[i][Mod(ip,3)+1];
                ib = mtj[i][Mod(ip+1,3)+1];
                ic = mtj[i][ip];
                jacia = jac[i][Mod(ip,3)+1];
                jacib = jac[i][Mod(ip+1,3)+1];
                mtj[i][1]= ia;
                mtj[i][2]= ib;
                mtj[i][3]= ic;
                mtj[i][4]= mtj[jn][ie3];
                jac[i][1]= jacia;
                jac[i][2]= jacib;
                inn1 = jac[jn][ie2];
                inn2 = jac[jn][ie3];

                if(inn1!=0){quedge(inn1,jn,jac,ien1);    jac[inn1][ien1]= i;}
                if(inn2!=0){quedge(inn2,jn,jac,ien2);    jac[inn2][ien2]= i;}
```

```
            jac[i][3]= inn1;
            jac[i][4]= inn2;

            for(k=1;k<=3;k++){
                mtj[jn][k]= 0;
                jac[jn][k]= 0;
                idm[jn]= 0;}

            iv = iv + 1;
            kv[iv]= jn;
        }//i

        qudelete(nelm,mtj,jac,idm,iv,kv,map);
}       //quexelm

public void qudelete(int nelm,int [][]mtj,int [][]jac,int []idm,int iv,int []kv,int [] map){
        m = 0;
        n = 0;
        for(i=1;i<=nelm;i++){map[i]=1; }
        for(i=1;i<=iv;i++)    {map[kv[i]] = 0;}
        for(i=1;i<=nelm;i++){ if(map[i]!=0){m = m + 1;    map[i]= m;}       }

        for(i=1;i<=nelm;i++){
            if(map[i]!=0){
                n = n + 1;
                for(ia=1;ia<=4;ia++){
                    mtj[n][ia]= mtj[i][ia];
                    idm[n]= idm[i];
                    if(jac[i][ia]== 0) jac[n][ia]= 0;
                    if(jac[i][ia]!= 0) jac[n][ia]= map[jac[i][ia]];}//ia
            }     }    //i

        for(i=n+1;i<=nelm;i++){
                for(ia=1;ia<=4;ia++){
                mtj[i][ia]= 0;
                jac[i][ia]= 0;
                idm[i] = 0;}
        }//i
```

```
            nelm = nelm - iv;
}     //qudelete


public void quisogen(int node,int nelm,int [][]mtj,int [][]jac,double []px,double
[]py,int []id,int []mmtj, int []ifix,int []idm){

            for(i=1;i<=nelm;i++){
                  if(mtj[i][4]== 0){
                        id[i]= 3;
                        mtj[i][5]= mtj[i][3];
                        mtj[i][3]= mtj[i][2];
                        mtj[i][2]= 0;}
                  if(mtj[i][4]!= 0){
                        id[i]= 4;
                        mtj[i][7]= mtj[i][4];
                        mtj[i][5]= mtj[i][3];
                        mtj[i][4]= 0;
                        mtj[i][3]= mtj[i][2];
                        mtj[i][2]= 0;}
            } //i

            for(i=1;i<=nelm;i++){
                  for(j=1;j<=id[i];j++){
                        if(mtj[i][2*j]!=0) continue;
                        node = node + 1;
                        mtj[i][2*j]=node;
                        px[node]= (px[mtj[i][2*j-1]] + px[mtj[i][Mod(2*j,2*id[i]+1)]]) /
2.0;
                        py[node]= (py[mtj[i][2*j-1]] + py[mtj[i][Mod(2*j,2*id[i]+1)]]) /
2.0;
                        if(jac[i][j]== 0){ ifix[node]= 1; continue; }
                        jn = jac[i][j];
                        quedge(jn, i, jac, ien);
                        if(mtj[jn][2*ien]!=0) continue;
                        mtj[jn][2*ien]= node;
                        if(idm[i]!=idm[jn]) ifix[node] = 1;}    //j
```

```
                node = node + 1;
                mtj[i][2*id[i]+1]= node;
                pxx = 0.0;
                pyy = 0.0;
                for(k =1;k<=id[i];k++){
                      pxx = pxx + px[mtj[i][2*k]];
                      pyy = pyy + py[mtj[i][2*k]];}

                px[node]= pxx /(double)id[i];
                py[node]= pyy /(double)id[i];      } //i

          qusquare(node,nelm,mtj,px,py,id,mmtj);
}   //quisogen

public void changemtj(int nelm,int [][]mtj,double []px,double []py){
      double rmin,r;
      int i,m,mm;
      mm=0;
      double []ppx=new double[4];
      double []ppy=new double[4];

      for(i=1;i<=nelm;i++){

          for(m=1;m<=4;m++){ppx[m]=px[mtj[i][m]];ppy[m]=py[mtj[i][m]];}

          rmin=Math.sqrt(ppx[1]*ppx[1]+ppy[1]*ppy[1]);

          for(m=2;m<=4;m++){
              r=Math.sqrt(ppx[m]*ppx[m]+ppy[m]*ppy[m]);
              if(r<rmin) {rmin=r; mm=m;}
          }
          if(mm==2){
              px[mtj[i][1]]=ppx[2]; py[mtj[i][1]]=ppy[2];
              px[mtj[i][2]]=ppx[3]; py[mtj[i][2]]=ppy[3];
              px[mtj[i][3]]=ppx[4]; py[mtj[i][3]]=ppy[4];
              px[mtj[i][4]]=ppx[1]; py[mtj[i][4]]=ppy[1];}
          if(mm==3){
```

```
            px[mtj[i][1]]=ppx[3]; py[mtj[i][1]]=ppy[3];
            px[mtj[i][2]]=ppx[4]; py[mtj[i][2]]=ppy[4];
            px[mtj[i][3]]=ppx[1]; py[mtj[i][3]]=ppy[1];
            px[mtj[i][4]]=ppx[2]; py[mtj[i][4]]=ppy[2];}
        if(mm==4){
            px[mtj[i][1]]=ppx[4]; py[mtj[i][1]]=ppy[4];
            px[mtj[i][2]]=ppx[1]; py[mtj[i][2]]=ppy[1];
            px[mtj[i][3]]=ppx[2]; py[mtj[i][3]]=ppy[2];
            px[mtj[i][4]]=ppx[3]; py[mtj[i][4]]=ppy[3];}

    }

}//changemtj
public void quedge(int l,int k,int [][]jac,int iedge){

        iedge = -1;
        for(i=1;i<=4;i++){if(jac[l][i]== k){ iedge = i;break;} }
        if(iedge== -1){ tf_command.setText("發生錯誤中止執行 edge");    }
} //quedge

public void qusquare(int node,int nelm,int [][]mtj,double []px,double []py,int []id,int
[][]mmtj){

        nelmm = 0;
        for(i= 1;i<=nelm;i++){
            if(mtj[i][9]==0){
                for(j=1;j<=id[i];j++){
                    nelmm = nelmm + 1;
                    mmtj[nelmm][1]= mtj[i][Mod(2*j+3,6)+1];
                    mmtj[nelmm][2]= mtj[i][2*j-1];
                    mmtj[nelmm][3]= mtj[i][2*j];
                    mmtj[nelmm][4]= mtj[i][7];
            }}
            if(mtj[i][9]!=0){
                for(j=1;j<=id[i];j++){
                    nelmm = nelmm + 1;
                    mmtj[nelmm][1]= mtj[i][Mod(2*j+5,8)+1];
                    mmtj[nelmm][2]= mtj[i][2*j-1];
```

```
                        mmtj[nelmm][3]= mtj[i][2*j];
                        mmtj[nelmm][4]= mtj[i][9];
                }}
            }
             nelm = nelmm;
}     //qusquare

public void SquareRelocation(){
            sqlaplas(nelm1,node1,mtj1,px1,py1,ifix,jac1,jnb1,nei1);

            SquareWriteData();
}

public void sqlaplas(int nelm,int node,int [][]mtj,double []px,double []py,int []ifix,int
[][]jac,int []jnb,int [][]nei){


            for(i=1;i<=kte;i++){ for(j=1;j<=4;j++){ jac[i][j]=0;}}

            ix = 0;
            for(i=1;i<=ked;i++){
                  ihen[i][1]= 0;
                  ihen[i][2]= 0;
                  jhen[i]= 0;
                  khen[i]= 0;      }

            for(i=1;i<=ktj;i++){
                  jnb[i]= 0;
                  for(j=1;j<=kcm;j++){ nei[i][j]= 0;} }

            for(j=1;j<=nelm;j++){
            for(k=1;k<=4;k++){
                        ka = mtj[j][k];
                        kb = mtj[j][Mod(k,4)+1];
                        for(l=1;l<=ix;l++){
                              la = ihen[l][1];
                              lb = ihen[l][2];
                              if (ka==lb&&kb==la) {
```

```
                                jac[j][k]= jhen[l];
                                jac[jhen[l]][khen[l]] = j;}
                        else {
                                ix = ix + 1;
                                ihen[ix][1]= ka;
                                ihen[ix][2]= kb;
                                jhen[ix]= j;
                                khen[ix]= k;}
                        }
                //continue sqlaplas50;
//sqlaplas70:
//                      jac[j][k]= jhen[l];
//                      jac[jhen[l]][khen[l]] = j;
//sqlaplas50:
        }} //k,j

        for(i=1;i<=nelm;i++){
        for(j=1;j<=4;j++){
                if(jac[i][j]== 0){
                        ja = mtj[i][j];
                        jb = mtj[i][Mod(j,4)+1];
                        ifix[ja]= 1;
                        ifix[jb]= 1; }
         }}

        for(i=1;i<=nelm;i++){
        for(j=1;j<=4;j++){
                ip = mtj[i][j];
                jnb[ip]= jnb[ip]+ 1;
                nei[ip][jnb[ip]]= i;    }}

        itera = 5;
        for(it=1;it<=itera;it++){
              for(i=1;i<=node;i++){
                if(ifix[i]== 0){
                        gx = 0.0;
                        gy = 0.0;
                        ar = 0.0;
```

```
                        for(j=1;j<=jnb[i];j++){
                            ielm = nei[i][j];
                            j1 = mtj[ielm][1];
                            j2 = mtj[ielm][2];
                            j3 = mtj[ielm][3];
                            j4 = mtj[ielm][4];
                            a1=.5*(px[j1]*py[j2]+px[j2]*py[j3]+px[j3]*py[j1]-
px[j1]*py[j3]-px[j2]*py[j1]-px[j3]*py[j2]);
                            a2=.5*(px[j1]*py[j3]+px[j3]*py[j4]+px[j4]*py[j1]-
px[j1]*py[j4]-px[j3]*py[j1]-px[j4]*py[j3]);
                            s= a1 + a2;
                            xc4 = (px[j1]+ px[j2]+ px[j3]+ px[j4]) / 4.0;
                            yc4 = (py[j1]+ py[j2]+ py[j3]+ py[j4]) / 4.0;
                            ar = ar + s;
                            gx = gx + s * xc4;
                            gy = gy + s * yc4;}

                        cgrax = gx / ar;
                        cgray = gy / ar;
                        px[i]= cgrax;
                        py[i]= cgray;
                    }
            }}
}   //sqlaplas

public void    SquareWriteData(){

      AxisChange();

      try{ pw_squdata=new PrintWriter(new BufferedWriter(new
FileWriter(file_squdata)));

          pw_squdata.printf("%10d%10d%10d\n",pseudoN1,pseudoN2,pseudoN3);
          pw_squdata.printf("%10d\n",nin);
          if(nin>0){for(i=1;i<=nin;i++) {pw_squdata.printf("%10d\n",ibin[i]); } }

if(islandNumm>0){for(i=1;i<=islandNumm;i++){pw_squdata.printf("%10d%10d%10d\
n",islandStart[i],islandEnd[i],islandNum[i]);} }
```

```
pw_squdata.printf("%10d%10d%10d\n",rightcoastStart,rightcoastEnd,rightcoastNum
);
        pw_squdata.printf("%10d%10d%10d\n",quayStart,quayEnd,quayNum);

pw_squdata.printf("%10d%10d%10d\n",leftcoastStart,leftcoastEnd,leftcoastNum);

pw_squdata.printf("%10d%10d%10d\n",innNodeStart,innNodeEnd,innNodeNum);

pw_squdata.printf("%10d%10d%10d\n",innPointStart,innPointEnd,innPointNum);

        pw_squdata.printf("%10d%10d\n",node1,nelm1);
        pw_squdata.printf("%10d%10d\n",node2,nelm2);
        pw_squdata.printf("%10d%10d\n",node3,nelm3);
        pw_squdata.printf("%10d%10d\n",node4,nelm4);
        pw_squdata.printf("%10d%10d\n",node5,nelm5);
        pw_squdata.printf("%10d%10d\n",node6,nelm6);


for(i=1;i<=node1;i++){pw_squdata.printf("%10d%10d%10d%10d\n",jnb1[i],px1[i],py1
[i],pz1[i]);}

for(i=1;i<=node2;i++){pw_squdata.printf("%10d%10d%10d%10d\n",jnb2[i],px2[i],py2
[i],pz2[i]);}

for(i=1;i<=node3;i++){pw_squdata.printf("%10d%10d%10d%10d\n",jnb3[i],px3[i],py3
[i],pz3[i]);}

for(i=1;i<=node4;i++){pw_squdata.printf("%10d%10d%10d%10d\n",jnb4[i],px4[i],py4
[i],pz4[i]);}

for(i=1;i<=node5;i++){pw_squdata.printf("%10d%10d%10d%10d\n",jnb5[i],px5[i],py5
[i],pz5[i]);}

for(i=1;i<=node6;i++){pw_squdata.printf("%10d%10d%10d%10d\n",jnb6[i],px6[i],py6
[i],pz6[i]);}

        for(i=1;i<=node1;i++){
```

```
      pw_squdata.printf("%10d%10d%10d%10d%10d%10d%10d%10d\n",nei1[i][1],n
ei1[i][2],nei1[i][3],nei1[i][4],

nei1[i][5],nei1[i][6],nei1[i][7],nei1[i][8]);}
            for(i=1;i<=node2;i++){

      pw_squdata.printf("%10d%10d%10d%10d%10d%10d%10d%10d\n",nei2[i][1],n
ei2[i][2],nei2[i][3],nei2[i][4],

nei2[i][5],nei2[i][6],nei2[i][7],nei2[i][8]);}
            for(i=1;i<=node3;i++){

      pw_squdata.printf("%10d%10d%10d%10d%10d%10d%10d%10d\n",nei3[i][1],n
ei3[i][2],nei3[i][3],nei3[i][4],

nei3[i][5],nei3[i][6],nei3[i][7],nei3[i][8]);}
            for(i=1;i<=node4;i++){

      pw_squdata.printf("%10d%10d%10d%10d%10d%10d%10d%10d\n",nei4[i][1],n
ei4[i][2],nei4[i][3],nei4[i][4],

nei4[i][5],nei4[i][6],nei4[i][7],nei4[i][8]);}
            for(i=1;i<=node5;i++){

      pw_squdata.printf("%10d%10d%10d%10d%10d%10d%10d%10d\n",nei5[i][1],n
ei5[i][2],nei5[i][3],nei5[i][4],

nei5[i][5],nei5[i][6],nei5[i][7],nei5[i][8]);}
            for(i=1;i<=node6;i++){

      pw_squdata.printf("%10d%10d%10d%10d%10d%10d%10d%10d\n",nei6[i][1],n
ei6[i][2],nei6[i][3],nei6[i][4],

nei6[i][5],nei6[i][6],nei6[i][7],nei6[i][8]);}


for(i=1;i<=nelm1;i++){pw_squdata.printf("%10d%10d%10d%10d\n",mtj1[i][1],mtj1[i]
```

```
[2],mtj1[i][3],mtj1[i][4]);}

for(i=1;i<=nelm2;i++){pw_squdata.printf("%10d%10d%10d%10d\n",mtj2[i][1],mtj2[i]
[2],mtj2[i][3],mtj2[i][4]);}

for(i=1;i<=nelm3;i++){pw_squdata.printf("%10d%10d%10d%10d\n",mtj3[i][1],mtj3[i]
[2],mtj3[i][3],mtj3[i][4]);}

for(i=1;i<=nelm4;i++){pw_squdata.printf("%10d%10d%10d%10d\n",mtj4[i][1],mtj4[i]
[2],mtj4[i][3],mtj4[i][4]);}

for(i=1;i<=nelm5;i++){pw_squdata.printf("%10d%10d%10d%10d\n",mtj5[i][1],mtj5[i]
[2],mtj5[i][3],mtj5[i][4]);}

for(i=1;i<=nelm6;i++){pw_squdata.printf("%10d%10d%10d%10d\n",mtj6[i][1],mtj6[i]
[2],mtj6[i][3],mtj6[i][4]);}


for(i=1;i<=nelm1;i++){pw_squdata.printf("%10d%10d%10d%10d\n",xn1[i],yn1[i],zn1[
i],area1[i]);}

for(i=1;i<=nelm2;i++){pw_squdata.printf("%10d%10d%10d%10d\n",xn2[i],yn2[i],zn2[
i],area2[i]);}

for(i=1;i<=nelm3;i++){pw_squdata.printf("%10d%10d%10d%10d\n",xn3[i],yn3[i],zn3[
i],area3[i]);}

for(i=1;i<=nelm4;i++){pw_squdata.printf("%10d%10d%10d%10d\n",xn4[i],yn4[i],zn4[
i],area4[i]);}

for(i=1;i<=nelm5;i++){pw_squdata.printf("%10d%10d%10d%10d\n",xn5[i],yn5[i],zn5[
i],area5[i]);}

for(i=1;i<=nelm6;i++){pw_squdata.printf("%10d%10d%10d%10d\n",xn6[i],yn6[i],zn6[
i],area6[i]);}

      pw_squdata.close();}
      catch(Exception e) {tf_command1.setText("WriteData");}
```

```
}
public int Mod(int x,int y) {
        int mod;
        mod=x-(x/y)*y;
        return mod;}

public void AxisChange(){
  for(i=1;i<=node;i++){
                AnyDepth((int)px1[i],(int)py1[i],z_xy) ;
                pz1[i]=z_xy;

              if(axis==1)    {
                  px1[i]=long_chart+                    px1[i]*xkgein /111000;
                  py1[i]=lati_chart+(crtgy*ykgein-py1[i]*ykgein)/111000;}
              if(axis==2)    {
                  px1[i]=long_chart+                    px1[i]*xkgein;
                  py1[i]=lati_chart+(crtgy*ykgein-py1[i]*ykgein);}
      }
}    //AxisChange

  public void ReadChart(){
                    dir    =tf_dir.getText().trim();
                    chart=tf_chart.getText().trim();

                    file_chart=dir+"/"+chart+".chart";
                    file_depth=dir+"/"+chart+".depth";
                    file_bound=dir+"/"+chart+".bound";
                    file_jpg    =dir+"/"+chart+".jpg";
                    file_gif    =dir+"/"+chart+".gif";

                    file_left      =dir+"/"+chart+"_left"      +".gif";
                    file_right    =dir+"/"+chart+"_right"    +".gif";
                    file_middle =dir+"/"+chart+"_middle" +".gif";
                    file_quay      =dir+"/"+chart+"_quay"      +".gif";
                    file_surface=dir+"/"+chart+"_surface"+".gif";
                    file_seabed =dir+"/"+chart+"_seabed" +".gif";
```

```
            file_tridata =dir+"/"+chart+".tridata";
            file_squdata =dir+"/"+chart+".squdata";

            try{
                br_chart=new BufferedReader(new FileReader(file_chart));
                str=br_chart.readLine();
                str=br_chart.readLine();
                str=br_chart.readLine(); mscale       =Double.parseDouble(str);
                str=br_chart.readLine(); axis         =Integer.parseInt(str);
                str=br_chart.readLine(); long_chart=Double.parseDouble(str);
                str=br_chart.readLine(); lati_chart=Double.parseDouble(str);
                str=br_chart.readLine(); long_diag =Double.parseDouble(str);
                str=br_chart.readLine(); lati_diag =Double.parseDouble(str);
                str=br_chart.readLine(); ni_depth    =Integer.parseInt(str);
                str=br_chart.readLine(); nj_depth    =Integer.parseInt(str);
                str=br_chart.readLine(); crtgx        =Integer.parseInt(str);
                str=br_chart.readLine(); crtgy        =Integer.parseInt(str);
                str=br_chart.readLine(); eps_num0    =Integer.parseInt(str);
                 br_chart.close();}
                 catch(Exception e) {tf_command1.setText("chart");}
    }   //readChart

public void ReadgridDepth(){

  try{
      br_depth=new BufferedReader(new FileReader(file_depth));
      for(i=0;i<=ni_depth;i++){                    //y 方向
      for(j=0;j<=nj_depth;j++){                    //x 方向
          str=br_depth.readLine();
          gridDepth[i][j]       =Double.parseDouble(str.substring(0,10));
          j                     =(int)Double.parseDouble(str.substring(10,15));

          i                     =(int)Double.parseDouble(str.substring(15,20));
           }}
    br_depth.close();}
    catch(Exception e) {tf_command.setText("depth1");}
}   // readDepth
```

```
  public void AnyDepth(int xc,int yc,double z_xy){
          double dx,dy,xr_a,yr_a,tx,ty,r1,r2;
          int ii,jj;
          r1=0;
          r2=0;
          dx = (crtgx / (double)nj_depth);
          dy = (crtgy / (double)ni_depth);
          jj =(int)(xc/dx);
          ii =(int)(yc/dy);
          xr_a = xc-jj*dx;
          yr_a = yc-ii*dy;
          tx = xr_a / dx;
          ty = yr_a / dy;
          if(xr_a>=0){r1=gridDepth[ii][jj]    +tx*(gridDepth[ii][jj+1]    -gridDepth[ii][jj]);}
          if(xr_a< 0){r1=gridDepth[ii][jj-1]+tx*(gridDepth[ii][jj]        -gridDepth[ii][jj-1]);}
          if(yr_a>=0){r2=gridDepth[ii+1][jj]+tx*(gridDepth[ii+1][jj+1]-
gridDepth[ii+1][jj]);}
          if(yr_a< 0){r2=gridDepth[ii][jj]    +tx*(gridDepth[ii][jj+1]    -gridDepth[ii][jj]);}
          z_xy=r1+ty*(r2-r1);
}       //AnyDepth

public void End() {
        if(procedure==2) {rightcoastEnd=dki-1;              rightcoastNum=rightcoastEnd-
rightcoastStart+1;}
        if(procedure==3) {quayEnd=dki-1          ;            quayNum=quayEnd-
quayStart+1;}
        if(procedure==4) {leftcoastEnd=dki-1 ;              leftcoastNum =leftcoastEnd-
leftcoastStart+1;}
        if(procedure==5) {islandEnd[islandNumm]=dki-
1;islandNum[islandNumm]=islandEnd[islandNumm]-islandStart[islandNumm]+1;}
        if(procedure==6) {innNodeEnd=dki-1;                 innNodeNum=innNodeEnd-
innNodeStart+1;}
        if(procedure==7) {innPointEnd=dki-1;                innPointNum=innPointEnd-
innPointStart+1;}

        tempN=dki-1;
        }//end
```

```
public void All_End() {

        tf_command1.setText("存檔");
        tf_command.setText("存檔中") ;
        if(nin>0){
            for(i=1;i<=nin;i++) {ibin[i]=islandNum[i];}
            totalInn=0;
            for(i=1;i<=nin;i++) {totalInn=totalInn+ibin[i];}
        }

        nob=pseudoN+rightcoastNum+quayNum+leftcoastNum+totalInn;//外部及離島群邊界上節點總數
        nib=innNodeNum+innPointNum;                                    //內部節點總數
        ibex=pseudoN+rightcoastNum+quayNum+leftcoastNum;          //外部邊界上節點總數

        boundN1=nob+nib;
        totalN1=boundN1+innNodeNum+ innPointNum;

        try{ pw_bound=new PrintWriter(new BufferedWriter(new FileWriter(file_bound)));

            pw_bound.printf("%10d\n",node);
            pw_bound.printf("%10d\n",nin);
            pw_bound.printf("%10d\n",ibex);
            pw_bound.printf("%10d\n",nob);
            pw_bound.printf("%10d\n",nib);

            if(nin>0){for(i=1;i<=nin;i++) {pw_bound.printf("%10d\n",ibin[i]);}      }

for(i=1;i<totalN1;i++){pw_bound.printf("%10d%10d%10d%10.2f\n",ibno1[i],(int)px1[i],(int)py1[i],pz1[i]);}

            pw_bound.close();}
        catch(Exception e) {tf_command1.setText("bound");}

         tf_ibex.setText(Integer.toString(ibex));
```

```
                tf_nob.setText(Integer.toString(nob));
                tf_nib.setText(Integer.toString(nib));
                tf_command.setText("完成存檔") ;

                LeftBoundary()       ;
                MiddleBoundary();
                RightBoundary() ;
                Quay()                ;

} //all_end



} //all
```