第9章

色々なアルゴリズム

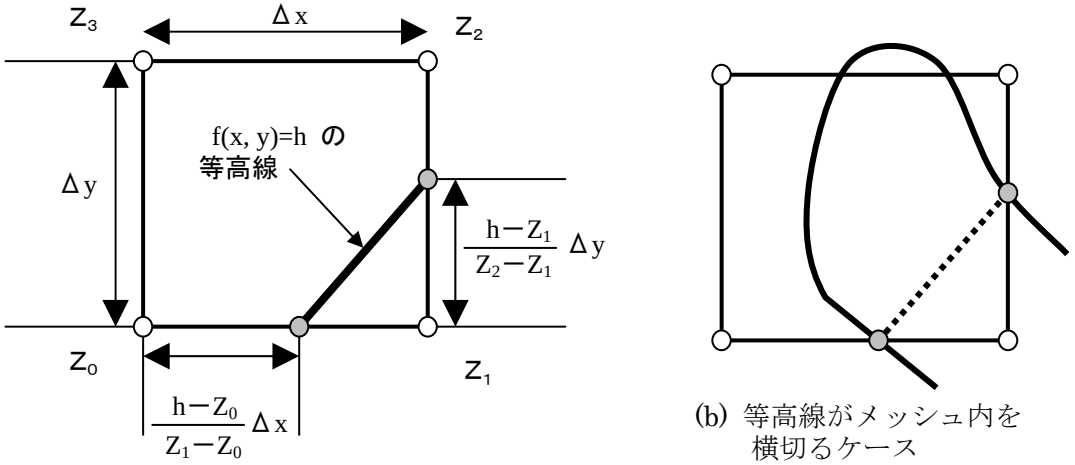
9.1　等高線

(1)考え方

等高線は，地形図等を表示する有力な方法です。ここでは，等高線を描く簡便な方法について示します。

通常の数値地図等はメッシュデータで表現されます。ここでは，横 Δx，縦 Δy の長方形の各辺に h となる点を見つけて，線を引きます。その計算方法を図 9-1(a)に示します。

ただし，図 9-1(b)のように交点が 3 つ以上あるような等高線，すなわち等高線がメッシュ内を横切るケースもあります。この場合，点線のように描くことになりますが，メッシュを十分小さくとれば問題ありません。



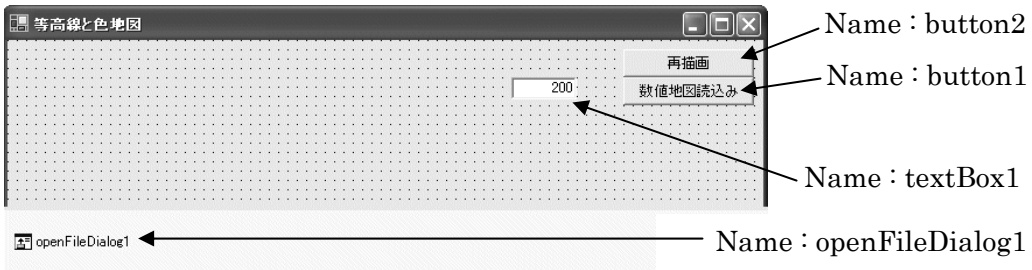
(a) 等高線の計算

(b) 等高線がメッシュ内を横切るケース

図 9-1　メッシュと等高線

(2)フォームの定義と実行例

次のようなフォームを定義しておきます。



提示するプログラムを実行すると，図 9-2(a)の画面になります。

右上の「数値地図読込み」ボタンをクリックして，国土地理院発行の 50 m メッシュ数値地図(いわゆる MEM ファイル)を指定すると，たとえば図 9-2(b)のように等高線が描かれます。なお。背景に色地図を表示します。

(a) 初期画面　　(b) 実行例

図 9-2　等高線を描くプログラムの実行画面

(3)プログラムの説明

2 次元の絵を描きます。また，数値地図のファイルを読み込みます。したがって，以下の 2 つの using を追加しておきます。

```
using System.Drawing.Drawing2D;
using System.IO;
```

【データ領域の宣言】

```
private int numX  = 201;  private int numY  = 201;
private double dx = 1.0;  private double dy = 1.0;
private double X0, Y0  ;  private string pbA;
private double Hstep;
private double[,] Z   = new double[201,201];
private Matrix matrix = new Matrix();    // 座標変換用データ
private Image image; // 描画用変数（この変数に描画します）
```

【OnPaint のオーバライド】
　ウインドウの描画が必要になると，Control クラスの OnPaint メソッドが呼び出されますので，これをオーバライドしておきます。

```
protected override void OnPaint(PaintEventArgs e)
{ base.OnPaint(e); e.Graphics.DrawImage(image,0,0);}
```

【補間のための関数】

```
public double 補間(double H, double Z1, double Z2)
{    return ((H-Z1)/(Z2-Z1));      }
```

【等高線を描く】

```
public void Contour(Graphics g, int j, int k, double H)
{ double Z0 = Z[j  , k  ]; double Z1 = Z[j+1, k ];
  double Z2 = Z[j+1, k+1]; double Z3 = Z[j  , k+1];
  bool P01 = (Z0> H) ^ (Z1 > H);  bool P12 = (Z1> H) ^ (Z2 > H);
  bool P23 = (Z2> H) ^ (Z3 > H);  bool P30 = (Z3> H) ^ (Z0 > H);
  if( P01 || P12 || P23 || P30)
  { float x0=0; float y1=0; float x2=0; float y3=0;
    Pen pen = new Pen(Color.Black,0.01F);
    if(P01) x0 =(float)( dx*(((double)j)+補間(H, Z0,Z1)));
    if(P12) y1 =(float)( dy*(((double)k)+補間(H, Z1,Z2)));
    if(P23) x2 =(float)( dx*(((double)j)+補間(H, Z3,Z2)));
    if(P30) y3 =(float)( dy*(((double)k)+補間(H, Z0,Z3)));
    float yk =(float)(dy*(double)(k ));
    float yk1=(float)(dy*(double)(k+1));
    float xj =(float)(dx*(double)(j ));
    float xj1=(float)(dx*(double)(j+1));
    if(P01 && P12) g.DrawLine(pen, x0 , yk , xj1, y1 );
    if(P12 && P23) g.DrawLine(pen, xj1, y1 , x2 , yk1);
    if(P23 && P30) g.DrawLine(pen, x2 , yk1, xj , y3 );
    if(P30 && P01) g.DrawLine(pen, xj , y3 , x0 , yk );
    if(P01 && P23) g.DrawLine(pen, x0 , yk , x2 , yk1);
    if(P12 && P30) g.DrawLine(pen, xj1, y1 , xj , y3 );
  }
}
public void Contours(Graphics g, double v1, double v2, double dv)
{ for(int j=0; j<numX-1; j++)
     for(int k=0; k<numY-1; k++)
        for(double v=v1; v<=v2; v +=dv) Contour(g,j,k,v);
 }
```

【色地図表示】

```
public void ColorMap(Graphics g, double v1, double v2, double dv)
{
  Color color;
  float w   = (float) dx;
  float h   = (float) dy;
  double DD = (v2-v1)/768;
  for(int j=0; j<numX-1; j++)
    for(int k=0; k<numY-1; k++)
    { float x1 =(float)( dx*((double)(j)));
      float y1 =(float)( dy*((double)(k)));
      int ID=(int)((Z[j,k]-v1)/DD);
      if      (ID<256) color=Color.FromArgb(    0    ,    ID    , 0);
      else if (ID<512) color=Color.FromArgb( ID - 256,   255   , 0);
      else if (ID<768) color=Color.FromArgb(   255   , 767 - ID, 0);
      else             color=Color.FromArgb(   255   ,    0    , 0);
      Brush brush = new SolidBrush(color);
      g.FillRectangle(brush,x1,y1,w,h);
    }
  }
```

【描画】

```
private void 描画()
{ image       = new Bitmap(1000,1000);
  Graphics g  = Graphics.FromImage(image);
  g.Clear(this.BackColor);
  g.Transform = matrix;
  double V1   = minZ();
  double V2   = maxZ();
  if(Math.Abs(V2-V1) < 0.0000001) V2 = V1 + 1;
  ColorMap(g, V1, V2, 0.1);
  Contours(g, V1, V2, Hstep);
}
```

【座標変換マトリックスの設定】

```
private void window(double X1, double Y1, double X2, double Y2)
{ double W = this.ClientSize.Width;
  double H = this.ClientSize.Height;
  float SX = (float)(W/(X2-X1))   ;
  float SY = (float)(H/(Y2-Y1));
  matrix.Scale(SX,SY) ;
  matrix.Translate(-(float)X1,-(float)Y1);
}
```

【標高値の最大・最小を求める】

```
public double minZ()
{
  double R=Z[0,0];
  for(int j=0; j<numX-1; j++)
     for(int k=0; k<numY-1; k++) if(Z[j,k]<R) R=Z[j,k];
  return R;
}
public double maxZ()
{
  double R=Z[0,0];
  for(int j=0; j<numX-1; j++)
     for(int k=0; k<numY-1; k++) if(Z[j,k]>R) R=Z[j,k];
  return R;
}
```

【初期処理】

```
private void Form1_Load(object sender, System.EventArgs e)
{
  double XX = ((double) numX) * 0.5;
  double YY = ((double) numY) * 0.5;
  X0        = dx * XX;
  Y0        = dy * YY;
  double C  = 0.02; //倍率
  Hstep=double.Parse(textBox1.Text);
  for(int i = 0; i < numX; i++)
     for(int j = 0; j < numY; j++)
     {
       double x = C * (((double)i) - XX);
       double y = C * (((double)j) - YY);
       Z[i,j]   = (x - y) * Math.Exp( -(x * x + y * y)) * 5000;
     }
  window(-20,240,220,-50); 描画();
}
```

【「ファイルを開く」ダイアログの表示】

```
private void button1_Click(object sender, System.EventArgs e)
{
  openFileDialog1.ShowDialog();
}
```

【数値地図を読み込んで描画】

```
private string midStr(string DT, int ist,int N)
{ string S=""; int k=ist-1;
  for(int i=1 ;i<=N;i++) S += DT[k++];
  return S;
}
private void openFileDialog1_FileOk
            (object sender, System.ComponentModel.CancelEventArgs e)
{
  StreamReader DTS; int ii, pbll;
  string FName = openFileDialog1.FileName;
  if(FName == "") return;
  DTS       = new StreamReader(FName,System.Text.Encoding.Default);
  pbA       = DTS.ReadLine();
  pbll      = 0;
  string DT = DTS.ReadLine();
  while(DT !=null)
  { ii = 5;
    for(int j = 0;j < 200; j++)
    {  ii += 5;  Z[pbll, j] = int.Parse(midStr(DT, ii, 5));
    }
    DT=DTS.ReadLine(); pbll++;
  }
  DTS.Close();  描画();
  this.Invalidate();
}
```

【テキスト変更時の等高線間隔を変更】

```
private void textBox1_TextChanged(object sender, System.EventArgs e)
{ Hstep=double.Parse(textBox1.Text);}
```

【button2 のイベントハンドラ(再描画)】

```
private void button2_Click(object sender, System.EventArgs e)
{  描画();  this.Invalidate();  }
```