

文章编号:1000-582x(2001)01-0089-03

等值线图形的扫描线法^{*}

袁政强, 祝家麟, 白绍良

(重庆大学B区建筑工程学院, 重庆 400045)

摘要:一般的等值线图采用函数值跟踪算法, 该法对内部封闭等值线的起点确定比较麻烦。笔者采用面向对象的程序设计方法给出适用于任意形状单元的等值线图的通用扫描线生成方法, 采用VC 6.0中的MFC实现了三点三角形、六点三角形、四点四边形、八点四边形和九点四边形的计算结果图形显示。并给出了算例。

关键词:面向对象; 计算机图形学; 等值线图; 数据可视化

中图分类号: TP 311.11; TP 391.41

文献标识码: A

等值线图是有限元分析过程中和地图处理过程中常用的图形。等值线是一种数据可视显示方法, 它利用人的视觉和计算机, 根据数据产生的图像, 利用颜色、密度、透明度、文字等技术, 将不可见的数字变为可见, 在短时间内通过图形传递给人脑大量信息, 使研究人员能够直观、迅速地观察到计算模拟的结果。它能直观地给出某一点的函数值。等值线图一般由区域内离散点上等位函数值通过插值函数插值得到任意点上等位函数值。线性插值函数有三点三角形、四点四边形。二次插值函数有六点三角形, 八点四边形和九点四边形。还有精度更高的高次插值函数。

一般的等值线图采用函数值跟踪算法, 即沿某一函数值求等值线等位走向。内部封闭等值线等位起点确定麻烦^[1]。笔者采用扫描线法, 能方便地确定等值线的坐标。扫描线算法是将函数值作为平面外的Z坐标, 离散的函数值用插值函数生成空间的曲面。用垂直于Z坐标的平截面去截空间曲面, 得到的在XY平面内的投影就是等值线。投影的计算是采用扫描线法, 即在平截面内给出许多平行的扫描线, 扫描线与空间曲面的交点在XY平面的坐标就是投影点。

面向对象的程序设计方法^[2,3]是将数据和处理数据的函数作为一个整体对象, 如所用到的点就可作为对象。将点用链表连接组成点集合^[4]。将插值函数应用的区域作为单元, 先定义一个抽象的单元, 不同区域用各个具体的单元继承抽象单元生成。所有单元用

链表连接。投影函数对于不同的单元是不同的。只要在抽象单元类中增加一个处理数据的函数接口, 在具体单元中增加处理函数即可。

1 等值线图的色谱类

本软件采用不同的颜色来绘制不同的等值线。对于一个函数值有一个且仅有一个颜色与之对应, 即为下面要给出的色谱。先求出函数的最大最小值, 将最小值对应一个BGR值Color1, 即三个整数代表的颜色值。最大值对应于另一个BGR值Color2, 一般函数值对应的颜色根据该函数值与最大最小值的比例关系由Color1与Color2的插值得到。色谱类如下:

```
class chromatogram{
public:
    int r, g, b;
    float Max, Min;
    chromatogram(float max, float min) : Max = max; Min = min; {}
    void Set(float c);
};

int Colorx[7][3] = {{255, 0, 0}, {0, 255, 0}, {0, 0, 255}, {255, 255, 0}, {255, 0, 255}, {0, 255, 255}, {0, 0, 0}};

void chromatogram::Set(float c){
    int n = 5, i;
    float d = (Max - Min) / n;
    for(i = 0; i < n; i++) if(i * d <= c && c < (i + 1) * d)
```

* 收稿日期: 2000-07-12

作者简介: 袁政强(1962-), 男, 贵阳人, 重庆大学副研究员, 在职博士生。主要从事计算数学和钢筋混凝土结构分析。

```
break;
d = (c - i * d) / d;
r = (1 - d) * Colorx[i][0] + d * Colorx[i + 1][0];
g = (1 - d) * Colorx[i][1] + d * Colorx[i + 1][1];
b = (1 - d) * Colorx[i][2] + d * Colorx[i + 1][2];
```

```
//是 Y 坐标,返回时是 x 投影
pDC->SetPixel((int)x,(int)y,RGB(col.r,
col.g,col.b));
};
};
```

色谱类的 MAX 和 MIN 是标量的最大最小值。Set(float c)函数是根据标量 c 求出该标量值对应的颜色 {r g b}。

2 等值线图的投影值计算

由于扫描线是在平截面内,平截面是垂直于 Z 轴的平面,扫描线的 Z 坐标与平截面的 Z 坐标相等,都假设为 Z_a 。假设扫描线由平行于 X 坐标的一族直线组成。扫描线的 Y 坐标根据图形的范围确定开始和结束的扫描线值。即先求出结点的最大 Y 坐标 MAX_y 和最小 Y 坐标 MIN_y ,第 1 条扫描线的 Y 坐标为: $Y = MIN_y$,第 2 条是 $Y + = \Delta Y$ 。最后一条是 $Y = MAX_y$ 。计算函数如下:

```
void Graph::ColorGraph (CDC * pDC)
{
    Point * p;
    float MAX_y = -2000., MIN_y = 2000., x, y;
    float Max_c = -2000., Min_c = 2000., c;
    char s[20];
    Path * ee;
    int l, i = 1;
    ListIterator<Point * > li( * Ps);
    while(li.NextNotNull())
    //计算坐标点的最大最小 y 值和函数值
        p = li.Next(); x = p->GetX(); y = p->GetY();
        c = p->Getc();
        if(y < MIN_y) MIN_y = y;
        if(y > MAX_y) MAX_y = y;
        if(c < Min_c) Min_c = c;
        if(c > Max_c) Max_c = c;
};
chromatogram col(Max_c, Min_c); //生成色谱对象
ListIterator<Path * > lei( * Es);
for(c = Min_c; c < Max_c; c += dC) // c 为等值线的值,既 Z
坐标值;dC 是等值线之间的差值
col.Set(c); // 根据函数值求颜色
for(y = MIN_y; y < MAX_y; y += 1.) // y 为扫描线的 y 值
    lei.First(); i = 1;
    while(lei.NextNotNull())
        ee = lei.Next(); //取单元
        x = y;
        if(ee->GetPixel(x,c) // 参数 x 进入时
```

扫描线空间曲面交点的确定:
离散函数值通过插值函数插值后得到的曲面的参数方程是:

$$x = \sum_{i=1}^n N_i(\xi, \eta) x_i \tag{1}$$

$$y = \sum_{i=1}^n N_i(\xi, \eta) y_i \tag{2}$$

$$z = \sum_{i=1}^n N_i(\xi, \eta) z_i \tag{3}$$

扫描线上点的坐标为 (X_a, Y_a, Z_a) 。扫描线与曲面的交点的坐标由以下方程确定:

$$x_a = \sum_{i=1}^n N_i(\xi, \eta) x_i \tag{4}$$

$$y_a = \sum_{i=1}^n N_i(\xi, \eta) y_i \tag{5}$$

$$z_a = \sum_{i=1}^n N_i(\xi, \eta) z_i \tag{6}$$

扫描线上 Y、Z 坐标为已知,可以通过方程(5)(6)求参数 (ξ, η) ,方程(5)(6)可变为以下形式:

$$f_1(\xi, \eta) = \sum_{i=1}^n N_i(\xi, \eta) y_i - y_a \tag{7}$$

$$f_2(\xi, \eta) = \sum_{i=1}^n N_i(\xi, \eta) z_i - z_a \tag{8}$$

得到的是非线性方程组,要用 Newton 迭代法求解。Newton 迭代法的切线矩阵为:

$$R = \begin{bmatrix} f_{1\xi} & f_{1\eta} \\ f_{2\xi} & f_{2\eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n N_{i\xi} y_i & \sum_{i=1}^n N_{i\eta} y_i \\ \sum_{i=1}^n N_{i\xi} z_i & \sum_{i=1}^n N_{i\eta} z_i \end{bmatrix} \tag{9}$$

迭代格式为:

$$\begin{bmatrix} \xi_{k+1} \\ \eta_{k+1} \end{bmatrix} = \begin{bmatrix} \xi_k \\ \eta_k \end{bmatrix} - R^{-1} \begin{bmatrix} f_1(\xi_k, \eta_k) \\ f_2(\xi_k, \eta_k) \end{bmatrix} \tag{10}$$

在求出点 (ξ, η) 后,落在区域内 X 坐标的 ξ, η 值首先要满足:

对四边形单元

$$-1 \leq \xi \leq 1, -1 \leq \eta \leq 1 \tag{11}$$
 或三角形单元

$$0 \leq \xi \leq 1, 0 \leq \eta \leq 1, 0 \leq 1 - \xi - \eta \leq 1 \quad (12)$$

X 的值用公式(4)计算。

以上是统一的计算公式,对于不同的单元要代入不同的形函数。形函数的公式可查^[5]。用面向对象的方法处理区域插值时,单元集合类的单元都是抽象单元类通过继承得到的具体单元的集合^[3],在抽象单元类中给出的虚函数 virtual int Getpixel(float &y, float &c)是计算交点的函数接口。返回的参数是表示曲面与扫描线是否相交。参数 y 进入时是扫描线 Y 值,参数 z 是扫描线的 Z 坐标。返回时参数 y 是交点的 x 坐标值。不同单元的计算只是 int Getpixel(float &y, float &c)的形函数和结点数不同。函数的流程完全相同。本软件给出了三点三角形单元、六点三角形单元、四点四边形单元、八点四边形单元和九点四边形单元的计算。向量模板定义见文献^[6]。下面是四结点四边形单元的计算函数:

```
int quad::Getpixel1(float &y, float &c)
{
    float d;
    Vector<float> xx(4), yy(4), cc(4);
    float a11, a12, a21, a22, dd;
    Vector<float> f(4), fn1(4), fn2(4), nn(2), n(2);
    int i, j;
    Point * p;
    i = 1;
    ListIterator<Point * > li(*El);
    while (li.NextNotNull())
    {
        p = li.Next(); //取第一结点
        xx(i) = p->Getx(); yy(i) = p->Gety();
```

```
        cc(i++) = p->Getc();
    };
    i = 1;
    while (1)
    {
        for (j = 1; j <= 4; j++)
        {
            f(j) = Shap2Dfunc0(4, j, n(1), n(2));
            fn1(j) = Shap2Dfunc1(4, j, n(1), n(2));
            fn2(j) = Shap2Dfunc2(4, j, n(1), n(2));
        }
        nn(1) = f * yy - y; // function 2
        nn(2) = f * cc - c; // function 1
        //Max model < 0.00001
        if (nn.MaxMo() < 0.00001 || i++ > 30) break
        a11 = fn1 * yy; a12 = fn2 * yy; //JACOBI Matrix
        a21 = fn1 * cc; a22 = fn2 * cc;
        dd = a11 * a22 - a12 * a21;
        n(1) = n(1) - (a22 * nn(1) - a12 * nn(2)) / dd;
        n(2) = n(2) - (a11 * nn(2) - a21 * nn(1)) / dd;
    };
    if (fabs(n(1)) <= 1 && fabs(n(2)) <= 1)
    {
        for (j = 1; j <= 4; j++)
            f(j) = Shap2Dfunc0(4, j, n(1), n(2));
        y = f * xx;
        return 1;
    }
    return 0;
};
```

3 算例

下面是方形区域内有三角形单元和四边形单元的

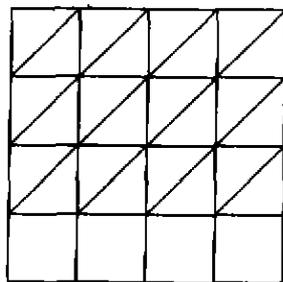


图 1 区域剖分图

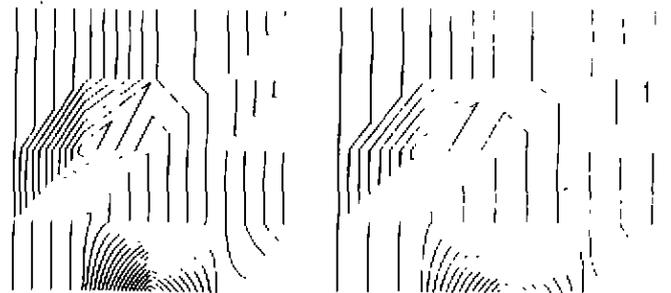


图 2 等值线图

计算算例,区域剖分如图 1,图 2 是不同密度的等值线计算结果如图。

对以上程序感兴趣的读者请与作者联系, E-mail:

zqyuan@public.cta.cq.cn

参考文献:

- [1] 江见鲸,贺小岗.工程结构计算机仿真分析[M].北京:清华大学出版社,1996
- [2] 谢春,马兰,彭颖红.有限元计算结果的可视化处理[J].计算机辅助设计与图形学学报,2000,12(2):81-84.

(下转 107 页)

