

一种新的等值线绘制方法研究

纛莹,朱航洲

(西安石油大学 计算机学院,陕西 西安 710065)

摘要: 提出了一种用 Surfer 软件提供的可编程方法对散点数据进行网格化,再用矩形追踪法追踪等值线的新方法。利用新方法绘制出的等值线既具有高精度性和实用性,又具有可维护性,这也表明该方法是一个实用、有效的等值线绘制方法。

关键词: Surfer; 等值线; 离散数据网格化; 矩形追踪

中图分类号: TP311

文献标识码: A

文章编号: 1674-7720(2012)03-0046-04

Study of a new method for contour-drawing

Cuan Ying, Zhu Hangzhou

(School of Computer Science, Xi'an Shiyou University, Xi'an 710065, China)

Abstract: A new drawing method has been proposed which uses Surfer software programmable approach to grid data and then use the rectangle tracing method to trace contour. The practice shows that it has high accuracy and practicality, furthermore it is maintainable and it is a practical and effective method.

Key words: Surfer; contour; the discrete data gridding; rectangular tracing

等值线图是在石油勘探开发、采矿、地质、地球物理、地球化学和气象等工程和技术领域内应用极广的一种图形,是众多领域成果表示的重要图件之一。在等值线绘制过程中,首先要对散点数据进行网格化,而常用的数据网格化方法(如距离加权法、方位法、趋势面法、加权最小二乘法、叠加法)的精度和有效性都不高;Kriging 估值技术虽然比传统的估值技术具有更高的先进性和有效性,但这种方法目前还没有得到广泛应用,许多细节问题还处于研究阶段。本文介绍利用 Surfer 软件提供的可编程方法对散点数据进行网格化,然后用矩形追踪法对网格化后的数据进行追踪,并实现了放大、缩小、漫游、光滑等功能,绘制出的等值线具有更强的实用性。

1 绘制等值线图原理

1.1 离散数据网格化

为了绘制等值线图,首先需要把离散分布的数据点网格化。为此,要建立数字高程模型(DEM),其主要功能就是将离散数据网格化。其主要步骤为:首先,由原始数据点的横、纵坐标的最小值和最大值构成矩形网格的边界;然后,根据用户需要将矩形再划分为 $m \times n$ 个小矩

形,即形成矩形网格;最后,用插值算法得到每个网格点的高程值。

DEM 内插有多种算法,常用的有距离加权法、样条函数内插、最小二乘配置法和 Kriging 方法等。本文使用 Surfer 软件中提供的可编程方法对散点数据网格化,其具体算法如下。

- (1) 输入参数;
- (2) 根据输入参数从数据库提取 X、Y 坐标及高程值 Z,并保存到文本数据文件;
- (3) 在程序中引用 surfer 并创建 surfer (srf) 和 grid (grd) 对象;
- (4) 用 srf 对象的 GridData 方法将文本文件转换成 surfer 格式文件(.grd 文件);
- (5) 用 grd 对象的 GetNode 方法获取 grd 文件里生成的网格点高程值,写入网格化后的数据文件。

1.2 等值点的计算

令任意一个网格边两端的数据值分别为 Z_1 、 Z_2 ,如图 1 所示,对于任意一条等值线 W ,它在该边上 $Z(x, y) - W = 0$,即等值线所通过的点,可以有 0~3 个根。

当等值线恰好等于网格节点的值时,这种网格点被

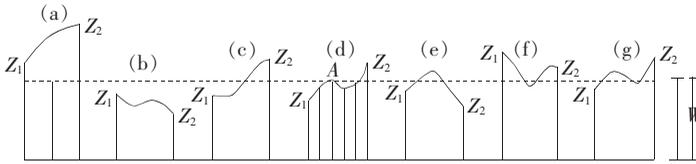


图1 网格边上存在等值线根的情况

称作退化点,当遇到退化点时,在网格点值上加上一个非常小的正数,以消除退化点^[3]。

如图1中的 Z_1 、 Z_2 值都不会等于等值线 W 的值,这时,网格边上对于某一条等值线 W ,存在根 $Z(x,y)-W=0$ 的可能性有:

(1) $Z_1 > W, Z_2 > W$ 或 $Z_1 < W, Z_2 < W$ 时,没有根,如图1中的(a)和(b),或存在两个根,如图1的(e)和(f)。

(2) $Z_1 < W, Z_2 > W$ 或 $Z_1 > W, Z_2 < W$ 时,存在一个根,如图1中的(c),或存在两个根,如图1中的(d),其中 A 根是重根,如图1中的(g)。

对于立方函数,利用数值法求根。这时可以将网格边长 n_x 和 n_y 再分成等间距的细分段,如图1中的(d)所示,然后求出每个细分段两端的函数值 Z_k 和 Z_{k+1} 。当 Z_k 或 Z_{k+1} 等于等值线值 W 时,同样需要加上一个非常小的正数,以消除退化点。这样,当 $(Z_k - W)(Z_{k+1} - W) < 0$ 时,说明该细分段上存在一个根。该细分段的长度可以小到以机器步长来计算,最小可以到描绘仪机器步长的2倍,这样可以保证在曲面函数上追踪所得到的等值线连续光滑而不会发生方向上的突变。在找到根值的细分段上,再以 Z_k 和 Z_{k+1} 作线性内插求得所在根及等值点的位置。

数值法找根时有时会失根,尤其对于重根情况,如图1(d)中的 A 根,由于它两端的 Z_k 和 Z_{k+1} 都小于 W ,因此 $(Z_k - W)(Z_{k+1} - W) > 0$, A 根就会失去,但这对于绘制等值线并无多大妨碍,至多漏绘一些极小的封闭等值线或等值线应该封闭而不完全封闭。

1.3 等值线的追踪

把网格纵边和横边的细分段所组成的小网格称为单元,它们的4个节点称为单元节点,以区别于原始的网格与网格节点。这样,对于每一个单元在其单元节点上求出函数值后(若遇到有退化点时,同样需要通过加一个很小的正数来处理),有图2所示的8种连接等值线的可能(其中+号表示节点值大于等值线值,-号表示节点值小于等值线值)。

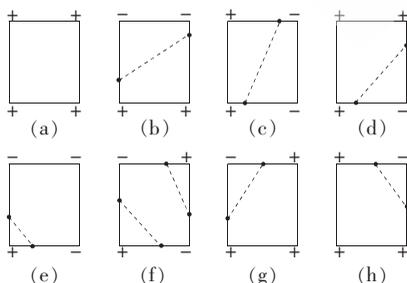
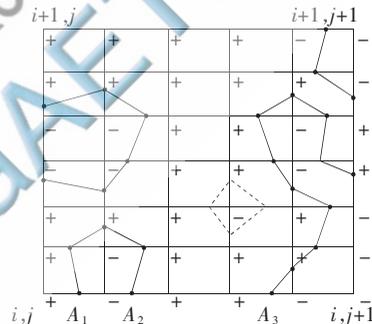


图2 单元内等值线的各种连接方法

(1) 单元节点上函数值同时大于或同时小于等值线值时,该单元内没有等值线通过,如图2(a)所示。

(2) 单元节点上函数值大于和小于等值线时,单元边上要么存在两个根,要么存在4个根。存在两个根时,两根在对边上,如图2(b)、图2(c)所示,两根在邻边上,如图2(d)、图2(e)、图2(g)、图2(h)所示;存在4个根时,4个根分别在单元的4个边上。对于两个根的情况,只要找到根的位置就可以相连;对于4个根的情况,则需规定底边根与左边根相连,右边根与上边根相连或底边根与右边根相连,右边根与上边根相连,如图2(f)所示,这样就能保证等值线间不会出现相交现象^[4]。

图3表示在一个矩形网格 (i,j) 内,将纵边 n_y 等分成6个细分段,将横边 n_x 等分成5个细分段的连法示意图。在等分后的每个单元节点上算出函数值 Z 。 $Z > W$ (W 为等值线值)时,以“+”表示; $Z < W$ 时,以“-”表示。这样在每个“+”、“-”节点之间必定存在一个等值线根,可用线性内插求出它的位置,连成的等值线如图3中的折线所示。由于网格边上的函数是立方函数,它在网格边上最多有3个根,如图3中,底边上有3个根 A_1 、 A_2 、 A_3 。在理论上只要对全部网格内的单元节点扫描一遍,把求得的根值用图2中的规则相连即可画出等值线图。

图3 网格 (i,j) 内分成单元后等值线连法示意图

但是这样计算会花费计算机大量时间,而且大部分单元内是不会有等值线通过的。因此,需要设计一种总等值线的方法,使得只在等值线会通过的单元节点上才求出函数值,而大部分不通过等值线的单元不必求节点函数值,以节省机器运算时间、加快速度。这里还需指出,图3是一个复杂化的例子,实际上,对于特定的等值线,一个网格内的等值线可能是很少的,或者只有一条或少数几条,还有很多网格根本就没有。此外,当分段加密时,等值线将逐渐变成光滑曲线,这是因为拟合网格点的函数是一个连续光滑的函数。因此,利用上述方法,就不需要对等值线再作光滑处理。

将等值线分为从边界出发到边界结束的等值线以及在内部封闭的等值线两种类型,如图4所示。无论哪种类型的等值线,都必须已知有顺序相邻的 A_1 、 A_2 两个等值点。设它们的坐标值分别为 (a_{1x}, a_{1y}) 及 (a_{2x}, a_{2y}) ,分别以 n_x 和 n_y 为单位长度,其坐标的整数值为 (a_{1j}, a_{1i}) 及 (a_{2j}, a_{2i}) 。任意设两个相邻的单元如图5中的单元I、II

所示, A_2 落在单元 I 和 II 的邻边上, A_1 落在单元 I 的其他三边的任意一边上。用 (i, j) 表示单元 I 的序号, 用 $(i+1, j)$ 、 $(i, j+1)$ 、 $(i-1, j)$ 、 $(i, j-1)$ 表示单元 II 的序号, 分别相当于图 5 中的 (a)、(b)、(c)、(d) 4 种情况, 则下一等值点追踪方向依次由下述 4 个判别式确定。

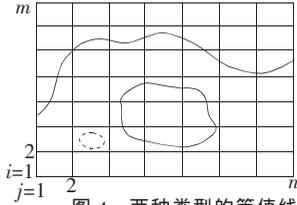


图 4 两种类型的等值线

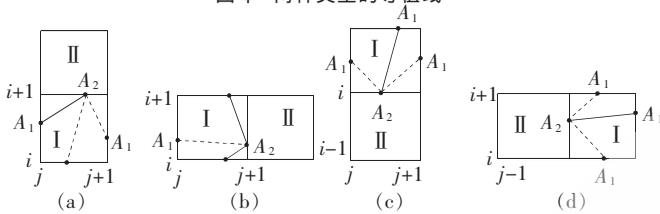


图 5 等值线追踪判别图

- (1) 若 $a_{1i} < a_{2i}$, 即图 5(a) 的情况, 下一点在单元 $(i+1, j)$ 的另外三边上;
- (2) 若 $a_{1j} < a_{2j}$, 即图 5(b) 的情况, 下一点在单元 $(i, j+1)$ 的另外三边上;
- (3) 若 $a_{2j} < a_{2i}$, 即图 5(c) 的情况, 下一点在单元 $(i-1, j)$ 的另外三边上;
- (4) 若 $a_{2i} < a_{2j}$, 即图 5(d) 的情况, 下一点在单元 $(i, j-1)$ 的另外三边上。

以图 4 的网格数组为例, 首先从网格数组四边寻找第一类等值点, 当在底边、左边、右边(或上边)的边界上找到一个等值点时, 令其为 A_2 。因为追踪一条等值线必须先要知道两个等值点, 所以必须再假设一个等值点 A_1 。当 A_2 在底边边界时, 令 A_1, A_2 的位置关系满足: $a_{1i} = -1, a_{1j} = a_{2j}$; 当 A_2 在左边界时, 令 $a_{1i} = a_{2i}, a_{1j} = -1$; 当 A_2 在右边(或上边)边界时, 令 $a_{1i} = a_{2i}, a_{1j} = a_{2j}$ 。然后利用 4 个判别式顺次进行判断, 即可追踪到下一等值点 A_3 。一旦追踪到 A_3 , 再把 A_2 充到 A_1 中, A_3 充到 A_2 中, 就可继续追踪, 一直追踪到图形边界为止, 这样就完成了第一类等值线的绘制。对于第二类等值线, 从内部网格的横边或纵边上先寻找它的第一个等值点, 令其为 A_2 。对于横边上的 A_2 , 再设一个等值点 A_1 , 使其和 A_2 的位置关系满足 $a_{1i} = -1, a_{1j} = a_{2j}$; 对于位于纵边上的 A_2 , 令其 $a_{1i} = a_{2i}, a_{1j} = -1$, 然后应用 4 个判别式顺次进行判断, 即可追踪到下一个等值点 A_3 。按照第一类等值点的追踪方法继续追踪下去, 直到重新追踪到首先找到的那个 A_2 点的位置为止, 这时便完成了一条封闭等值线的绘制。必须指出, 无论属于哪种类型的等值线, 每当追踪到等值点落在网格边上(而不是网格内部的单元边上)时, 必须将其位置记在一个表内。对于封闭性等值线, 还需把第一个等值点的位置记住, 以便每次开始找到第一个 A_2 点时,

与表中的位置作比较, 如果已经在表中, 则不再重绘。对于封闭等值线, 每追踪到一个位于网格上的点时, 还需同时记录下来的第一个等值点位置进行比较, 以判断是否回到原来的出发位置。

按上述方法追踪等值线, 可以避免计算许多没有等值线通过的单元的节点函数值, 但也有可能漏掉一些只在一个网格之内的小封闭等值线, 如图 4 中用虚线表示的等值线。但这些等值线在整个等值线图形中并不起重要作用, 忽略不绘并不影响图形质量, 却可以节省许多机器时间。

2 等值线绘制

关于等值线的绘制, 主要有数据处理和图形生成两部分。在数据处理部分, 主要是将离散数据网格化; 而在图形生成部分, 系统将采用矩形网格追踪法来生成等值线, 然后就要对所生成的等值线进行光滑处理和标注。

离散数据网格化主要调用 Surfer 提供的方法进行, 具体程序如下。

```
Public Sub LoadGrid (ByVal xyztxtname As String,
ByVal gridname As String, ByVal gridtxtname As String)
Dim srf As Object
Dim grd As Object
Dim a As Long
Dim b As Long
Dim Data() As Double
Dim zDev As Double
Dim zRoot As Double
Dim zSkew As Double
Dim zTrim As Double
Dim zMed() As Double
Filename=gridname
Set srf=CreateObject("surfer.application")
Set grd=srf.NewGrid
srf.GridData DataFile:=xyztxtname, ShowReport:=
False, outgrid:=gridname
grd.LoadFile gridname, False
xMin=grd.xMin
xMax=grd.xMax
xSize=grd.xSize
yMin=grd.yMin
yMax=grd.yMax
ySize=grd.ySize
zMin=grd.zMin
zMax=grd.zMax
NumRows=grd.NumRows
NumCols=grd.NumCols
zRange=zMax-zMin
zMidrange=(zMin+zMax)/2
```

```

BlankedNodes=0
FilledNodes=0
Dim fso, fsoorder
Set fso=CreateObject("scripting.filesystemobject")
Set fsoorder =fso.CreateTextFile (gridtxtname, True,
False)
Dim ssss
ssss=""
For a=1 To NumRows Step 1
For b=1 To NumCols Step 1
If grd.IsBlanked(a, b)=True Then
BlankedNodes=BlankedNodes + 1
Else
ReDim Preserve Data(FilledNodes)
Data(FilledNodes)=grd.GetNode(a, b)
ssss=ssss + CStr(Data(FilledNodes)) + " "
zSum=zSum + Data(FilledNodes)
FilledNodes=FilledNodes + 1
End If
Next b
Next a
fsoorder.Write "DSAA" + " " + Chr(13) + Chr
(10)
fsoorder.Write CStr(NumCols) + " " + CStr
(NumRows) + " " + Chr(13) + Chr(10)
fsoorder.Write CStr(xMin) + " " + CStr(xMax) +
" " + Chr(13) + Chr(10)
fsoorder.Write CStr(yMin) + " " + CStr(yMax) +
" " + Chr(13) + Chr(10)
fsoorder.Write CStr(zMin) + " " + CStr(zMax) +
" " + Chr(13) + Chr(10)
fsoorder.Write ssss
fsoorder.Close
srf.Quit
End Sub

```

网格追踪程序主要封装在 CContourTrace 类中。利用本文方法所生成的等值线如图 6 所示。本文是提出了一种新的方法来实现等值线的绘制,它首先用成熟软件 Surfer 将原始数据进行网格化,再利用矩形网格追踪法追踪网格,并按照实际需求实现了对

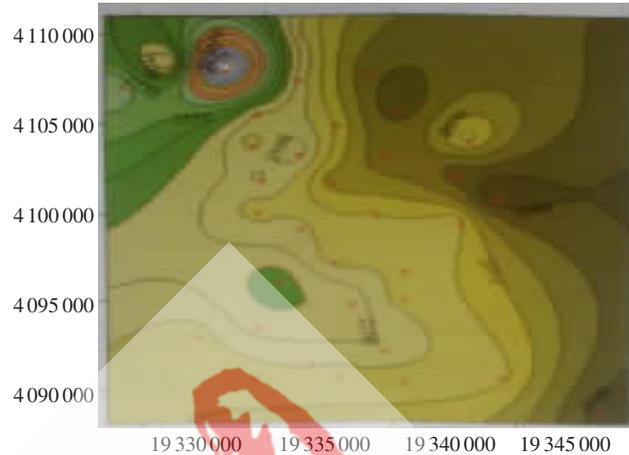


图 6 利用本文方法生成的等值线

等值线的缩放、平移、光滑和标注等功能。利用本方法所绘制的等值线主要在石油勘探开发、采矿等领域得到应用。使用本方法所绘制的等值图较其他方法效果更好、精确度更高、速度更快,是一个实用有效的等值线绘制方法。

参考文献

- [1] 王永会,宋晓宇,栾方军.基于网格的等值线图快速生成算法[J].计算机工程与应用,2001(17):124-125.
- [2] 陈学工,刘凯敏.一种基于格网法快速生成等值线的算法[J].电脑与信息技术,2007,15(3):4-6.
- [3] 宋丽娟,龚晓峰,钟猛.基于网格法的等值线绘制方法[J].现代电子技术,2005,28(14):65-67.
- [4] 韦美雁,杜丹蕾.基于规则网格的等值线的生成研究[J].湖南科技学院学报,2007,28(4):39-41.
- [5] 于嘉,吴旭.一种改进的矩形网格等值线追踪算法[J].河南师范大学学报,2008,36(6):34-36.
- [6] 张利,王俊彪,张贤杰.基于矩形网格追踪法的曲面主曲率等值线生成算法[J].计算机应用研究,2009,26(8):3179-3181.

(收稿日期:2011-11-24)

作者简介:

曩莹,女,1986年生,副教授,博士,主要研究方向:图像处理与模式识别。

朱航洲,男,1974年生,工程师,研究生,主要研究方向:图像处理与模式识别。