****

**研究生试卷**

**2018年—2019年度第二学期**

课 程 名 称： 电磁场数值模拟 评分： \_\_\_\_\_

专 业： 地球物理学 年级： 2018级

研 究 生 姓 名： 曹华科 学号：2018126002

任课教师姓名： 李貅 教授

**注意事项**

**1．答题必须写清题号；**

**2．字迹要清楚，保持卷面清洁；**

**3．试题随试卷交回；**

**4．考试课按百分制评分，考查课按5级分制评分；**

**5．阅完卷后，一周内将试卷、试题、成绩单由任课教师签名**

**后，送有关部门。**

**有限单元法在电磁场计算中的应用**

曹华科

（长安大学地质工程与测绘学院，西安 710054）

**摘要：**本文从《地球物理中的有限单元法》这本书出发，学习了变分法，从而建立了边值问题与泛函之间的关系，再引入形函数、里兹法和有限差分法结合，从而形成了如今的有限单元法。有限单元法从步骤上来，具有明显的特点：1.首先根据相关问题来建立边值问题；2.根据变分法和虚功原理（或最小位能原理）构建泛函，转化为变分问题；3.在单元内构造插值函数；4.进行单元分析，得到关于单元的小矩阵；5.单元合成，利用单元编号和整体编号，将各个单元的小矩阵合成一个大型的矩阵运算；6.线性方程组的求解。本文首先从书上位场延拓的程序进行学习，并用Fortran对其实现进行模拟实验，从而加深自己的理解。然后通过自己的理解，利用matlab对书上的电阻率电场（中间梯度法中间的均匀电场和点源二维电场）的有限单元法进行了编程实现，通过简单的网格剖分、程序调试和运行以及最后的结果分析，进一步加深了自己对有限单元法的理解，并且也证明了有限单元法在解决二维问题的有效性。最后利用三维有限单元法，研究了各种发射波形的正演，并利用视电阻率定义得到视电阻率图，从而分析了各种发射波形的成像特点。

**关键词：**二维正演，三维正演，任意发射波形

目录

[第一章 引言 1](#_Toc1876)

[第二章 位场延拓 2](#_Toc9172)

[2.1 变分问题 2](#_Toc4490)

[2.2 区域剖分 3](#_Toc31545)

[2.3 线性插值 4](#_Toc28773)

[2.4 单元分析 5](#_Toc23082)

[2.5 总体合成 6](#_Toc24876)

[2.6 求变分 10](#_Toc24229)

[2.7 解线性代数方程组 11](#_Toc20596)

[2.7 计算实例 13](#_Toc5960)

[第三章 电阻率模拟 16](#_Toc7477)

[3.1 二维均匀电场 16](#_Toc11938)

[3.1.1 问题 16](#_Toc14707)

[3.1.2 模型的建立 17](#_Toc8613)

[3.1.3 模型计算 18](#_Toc15695)

[3.2 点源二维电场 19](#_Toc25348)

[3.2.1 问题 19](#_Toc26489)

[3.2.2 傅里叶反变换 20](#_Toc6736)

[3.2.3 模型计算 22](#_Toc7768)

[第四章 直接时间域矢量有限元瞬变电磁三维正演 25](#_Toc25858)

[4.1磁场的双旋度方程 25](#_Toc24431)

[4.2 有限单元法 25](#_Toc28889)

[4.2.1 变分原理 25](#_Toc14007)

[4.2.2 网格剖分 26](#_Toc16954)

[4.2.3 单元分析 26](#_Toc26125)

[4.2.4 总体合成 27](#_Toc20079)

[4.3 回线源的加载 28](#_Toc25836)

[4.4 散度条件的施加 28](#_Toc4342)

[第五章 面向目标的自适应网格剖分 30](#_Toc25349)

[5.1 前沿 30](#_Toc15430)

[5.1.1 选题背景及研究意义 30](#_Toc3179)

[5.1.2 误差分析 30](#_Toc27054)

[5.2 后验误差估计的理论 31](#_Toc1149)

[5.2.1 控制网格细化的变量是一种后验误差值 31](#_Toc2364)

[5.2.2 面向目标的后验误差估计 31](#_Toc29867)

[5.2.3 自动网格加密的方法分类 32](#_Toc6606)

[5.2.4 面向目标的自动网格加密的流程 32](#_Toc14019)

[第六章 任意发射波形的视电阻率定义 34](#_Toc12987)

[6.1 视电阻率定义理论 34](#_Toc7802)

[6.2 三维计算模型 35](#_Toc29624)

[参考文献 38](#_Toc17468)

[附录 39](#_Toc21729)

# 第一章 引言

目前，电磁分布边值问题的数值计算方法包括有有限差分法、积分方程法、有限元法和边界单元法等四种基本类型，其中，有限元法占有绝对主要的地位，具有较大的应用范围。基于伽辽金或变分原理的有限元法(Finite Element Methods，简称FEM)将要分析的连续地球物理场分割为很多较小的区域，然后建立每个单元上待求场量的近似式，利用节点值与节点基函数形成整个求解区域电磁场的分布。从数学角度来讲，它是从变分原理出发，通过区域剖分和分片插值，把二次泛函的极值问题化为多元二次函数的极值问题，后者等价为求解一组多元线性代数方程组，是一种从部分到整体的方法，可使分析过程大为简化。

有限元法的出现，是数值分析方法研究领域重大的突破性进展，与其它数值方法相比较，有如下几方面的突出优点：

⑴通过单元分析，把二次泛函的极值问题等价于求解一组多元线性方程组的问题。这是一种从部分到整体的方法，可使分析过程大为简化；

⑵对于连续区域的离散，采用网格剖分，比较灵活，能较好的逼近不规则的地面和电性异常体，且易于按需要加密和放稀剖面网格，有利于实现以较少的计算量达到较高的计算精度；

⑶利用有限元法分析场问题，只要剖分处理得当，就可以得到较高的计算精度；

⑷有限元法可以成功地用于多种介质和非均匀连续介质问题，这是其他数值方法较难处理的问题，对于有限元却很容易经过简单的办法处理——只要对不同的单元规定不同的性质就可。多种介质和非均匀介质是地球物理介质的基本特征，因此有限元法的这个优点对物探来说是难得的；

⑸约束处理后的有限元方程系数矩阵是正定的，保证了解的存在唯一性，而且系数矩阵是稀疏的，可大大减少计算量和简化计算过程；

⑹解出各个节点值后，易于计算其区域内部场值。方法很有规则，易于在计算机上实现；

⑺有限元的发展、完善和应用与计算机技术的发展密切相关。近20年来计算机的运算速度和容量以惊人的速度提高，使有限元法的求解能力迅速增强。

# 第二章 位场延拓

**2.1 变分问题**

在地球物理中，磁源和引力源一般位于地下，在地表上部，没有场源。所以，位函数u应满足拉普拉斯方程：



然后来看位场延拓与变化中的边界条件。

通常只在地表测定u。不是一条封闭的边界，但边值问题要求，事先得知道整个闭合边界上的u。因此，我们必须加上一个无穷大边界，与组成封闭的边界。但上的u是未知的，我们用以下几种办法处理上的边值。

（1）将取得足够远，近似地认为。由于区域取得大，其中的节点数必然多，这就增加了计算工作量。

（2）将近似看作水平线，用水平线向上延拓的公式，由上的u计算出上的u，这样，不必取得很大，减轻了计算工作量，但上的u是近似的。

由以上二种方法，得到了第一类边界条件，其边值问题归结位：



其中代表已知函数。

（3）近似地认为地表的u是地下某处的集中的场源s产生的。在上，u的分别规律是



其中r是假想的源点s至边界上的点的距离，c是比例系数。通过变换，可以消去c，得到边界条件位：



此处，为了简单起见，选择了第一类边界条件作为边值问题：



根据最小位能原理：



对上面式子求变分，可得：



其中，由于，所以右边式子中的第二项0，只留下的第一项，利用斯托克斯定理，可得：



由于u在边界上为常数，所以在边界上恒为为0，所以为0，所以此时该泛函就是我们想要的变分问题：



## 2.2 区域剖分

用三角单元对整个区域进行剖分，因为三角单元的边容易拟合地形线的形状，矩形单元就没有这个优点。三角形的顶点称为节点，用节点上的离散的场值来近似场值的连续分布。单元的大小是使单元内的场值接近线性变化的原则，场值变化大的地方，单元小一些。单元的形状接近等边最好，不宜过锐或过钝。

剖分后，对节点和单元进行编号，将节点的 坐标和单元的节点序号列表（如下表），分别放在 和 两个二维数组中。 为节点总数， 为单元总数。

表1 节点的 坐标

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表1 节点的****坐标** |  |  |  |  |
| **节点号** | 1 | 2 | …… |  |
|  |  |  | …… |  |
|  |  |  | …… |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表2 单元的节点号** |  |  |  |  |
| **单元号** | 1 | 2 | …… |  |
|  |  |  | …… |  |
|  |  |  | …… |  |
|  |  |  | …… |  |

原则上，节点的编号是任意的，但是为了节省内存，要使同一单元上的节点号之差最小（其理由见本节总体合成）。单元上的节点号次序，即的次序，按逆时针方向排列。如果某单元的一个边落在区域的边界上，最好将边界上的节点作为和。

第一类边界条件上的场值是已知的，其余节点上的场值是待求的，所以要把第一类边界条件的节点号和节点上的场值 列表（如下表），分别放在 和 两个一维数组中。 为第一类边界条件的节点数。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表3 第一类边界条件的节点号和场值** |  |  |  |  |
| **序号** | 1 | 2 | …… |  |
| **第一类边界条件节点号** | …… | …… | …… | …… |
| **场值** |  |  |  |  |

## 2.3 线性插值

在三角单元内，假定场值是线性变化的

 （1）

而且，可以表示为

 （2）

其中，是形函数





## 2.4 单元分析

将全区域的积分分解为单元 的积分之和

 （3）

单元上的积分为

 （4）

其中是单元的值列向量；是单元系数矩阵，。因为，所以是对称矩阵。

计算单元系数矩阵的子程序如下：

（1）功能

三角单元/线性插值时计算单元系数矩阵

（2）程序

subroutine UKE1(X,Y,KE)

implicit none

real(kind=8):: X(3),Y(3),A(3),B(3)

real(kind=8) :: KE(3,3)

integer(kind=4) :: i,j

real(kind=8) :: S

A(1)=Y(2)-Y(3)

A(2)=Y(3)-Y(1)

A(3)=Y(1)-Y(2)

B(1)=X(3)-X(2)

B(2)=X(1)-X(3)

B(3)=X(2)-X(1)

S=2.\*(A(1)\*B(2)-A(2)\*B(1))

do i=1,3

do j=1,i

Ke(i,j)=( A(i)\*A(j)+B(i)\*B(j) ) / s

enddo

enddo

return

end subroutine

## 2.5 总体合成

将各单元的积分相加。相加前，将单元的场值扩展成全体节点的场值列向量:

 （5）

将的单元系数矩阵扩展成的单元系数矩阵



按照节点的总体序号，将单元系数矩阵中的各元，放在的相应行与列的交叉位置上，其余位置的元为零。这样，单元积分可写成

 （6）

因为各单元扩展后的列向量是相同的，所以各单元积分相加时，只要将相加即可：

 （7）

其中是总体系数矩阵。

在计算程序设计上，给出一个二维的总体系数矩阵数组，将各个单元系数矩阵中的元，加在总体系数矩阵数组中的相应位置即可。

总体系数矩阵由许多对称矩阵相加而成，所以也是对称的。此外在数学上可证明是正定的。为了节省计算机内存，只存储矩阵的下三角或是上三角不分。非零元素只存在于三角元三顶点编号所对应的行和列的九个交叉位置上，其他均为零元素，所以总体系数矩阵中包含大量零元素，成为稀疏矩阵。离主对角线最远的非零元素的位置取决于所有三角元顶点编号的最大差值的绝对值。称为系数矩阵的半带宽。通常，总体系数矩阵不是存放在的二维数组中，而是存放在 的二维数组中，称为定带宽储存。带的宽窄与否是节省内存的重要因素，所以在单元节点编号时，要尽可能使节点号之差最小。

下面给出求半带宽的子程序。

（1）功能

计算总体系数矩阵的半带宽。

（2）程序

subroutine MBW(NE,I3,IW)

implicit none

integer(kind=8) :: NE

integer(kind=8) :: IW,m

integer(kind=8) :: I3(3,NE)

integer(kind=4) :: i

IW=0

do i=1,NE

m=max(iabs(I3(1,i)-I3(2,i)),iabs(I3(2,i)-I3(3,i)),iabs(I3(3,I)-I3(1,I)))

if (m+1.GT.IW) then

IW=M+1

end if

end do

return

end subroutine

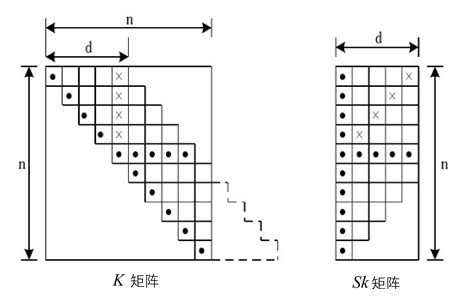
下面介绍定带宽存储总体系数矩阵的程序。

（1）定带宽存储思路

下图是的总体系数矩阵。由于矩阵是对称的，只画出下三角部分。“”代表非零元素，空白为零元素。图上矩阵的半带宽。将矩阵中的条带补充成平行四边形，再将它扳成矩形，此矩形是的二维数组，以作数组名。矩阵中的元素和数组中的元的关系为

 （8）

即将矩阵中的元素存储在数组中的中。



（2）功能

三角单元、线性插值时用定带宽存储的方法集成总体矩阵。

（3）程序

subroutine UK1(ND,NE,IW,I3,XY,SK)

implicit none

integer(kind=8) :: IW

integer(kind=8) :: NE,ND

integer(kind=8) :: I3(3,NE)

real(kind=8):: XY(2,ND),SK(ND,IW)

real(kind=8) :: X(3),Y(3)

real(kind=8) :: Ke(3,3)

integer(kind=4) :: NJ,NK

integer(kind=4) :: i,j,l,k

do i=1,ND

do j=1,IW

SK(i,j)=0

enddo

enddo

do L=1,NE

do j=1,3

i=I3(j,L) !取得对应单元中的节点号

x(j)=XY(1,i) !取得每个节点的x,y坐标

y(j)=XY(2,i)

enddo

call UKE1(X,Y,KE)

do j=1,3

NJ=I3(j,l)

do k=1,j

NK=I3(k,l) !获取Ke的双下标

if (NJ.LT.NK) then

NJ=NJ-NK+IW

SK(NK,NJ)=SK(NK,NJ)+KE(j,k)

NJ=NJ+NK-IW

else

NK=NK-NJ+IW

SK(NJ,NK)=SK(NJ,NK)+KE(J,K)

endif

enddo

enddo

enddo

return

end subroutine

## 2.6 求变分

通过以上四个步骤，已将连续函数 的泛函，离散成各节点值的多元函数：

 （9）

泛函的极值等于多元函数的极值，用多元函数求极值的方法，对上式求微分，

 （10）

其中，除第一类边界条件的节点的外，其余节点上的。因为是对称矩阵，有

 （11）

所以

 （12）

由于，所以由上式得

 （12）

这是含有个元的个方程联立的线性代数方程组。

## 2.7 解线性代数方程组

下面介绍代入第一类边界条件的子程序

（1）功能

在定带宽存储的总体系数矩阵和右侧列向量上加上第一类边界条件。

（2）程序

subroutine UB1(ND1,NB1,U1,ND,IW,SK,U)

implicit none

integer(kind=8) :: IW,ND1,ND

integer(kind=8) :: NB1(ND1)

real(kind=8):: U1(ND1),SK(ND,IW),U(ND)

integer(kind=4) :: i,j

do i=1,ND

u(i)=0

enddo

do i=1,ND1

j=NB1(i)

SK(j,IW)=SK(j,IW)\*1.e10

U(j)=SK(j,IW)\*U1(i)

enddo

return

end subroutine

代入第一类边界条件后，调用解定带宽存储的对称带型线性方程组的子程序解放程组。

（1）功能

对称带型线性方程组的系数矩阵的下三角部分被定带宽存储在矩形数组A中，利用A数组来解方程组。

（2）程序

subroutine LDLT(A,N,IW,P,IE)

integer(kind=8) :: N,IW

integer(kind=4) :: IE

real(kind=8) :: A(N,IW),P(N)

do 15 i=1,N

if(I.LE.IW) goto 20

IT=I-IW+1

goto 30

20 IT=1

30 K=I-1

IF(I.EQ.1)goto 40

do 25 L=IT,K

IL=L+IW-I

B=A(I,IL)

A(I,IL)=B/A(L,IW)

P(I)=P(I)-A(I,IL)\*P(L)

MI=L+1

do 25 j=MI,I

IJ=J+IW-I

JL=L+IW-J

25 A(I,IJ)=A(I,IJ)-A(J,JL)\*B

40 IF( A(I,IW).EQ.0. ) goto 100

15 continue

do 45 j=1,n

if( J.LE.IW ) goto 60

IT=N-J+IW

goto 70

60 IT=N

70 I=N-J+1

P(I)=P(I)/A(I,IW)

IF(J.EQ.1)goto 45

K=I+1

do 65 MJ=K,IT

IJ=I-MJ+IW

65 P(I)=P(I)-P(MJ)\*A(MJ,IJ)

45 continue

IE=0

goto 110

100 IE=1

110 return

end subroutine

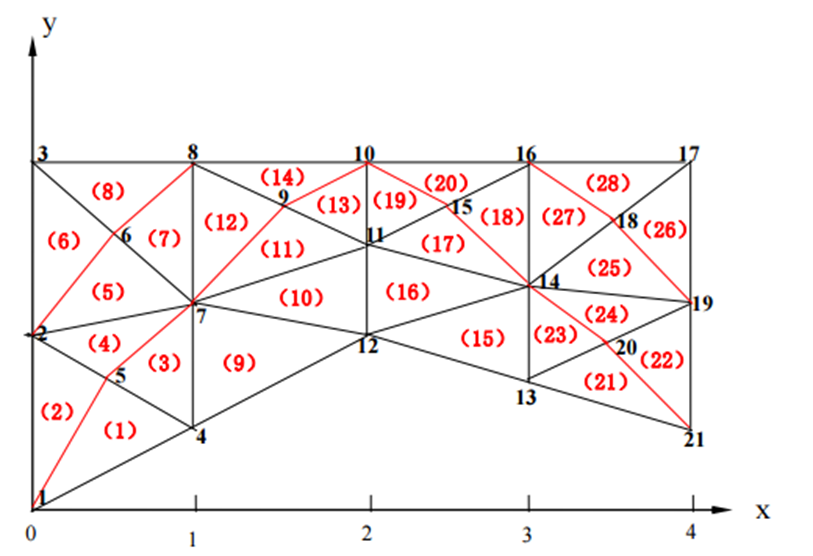
解代入第一类边界条件后的线性代数方程组 ，最后得到的各节点的 。至此，有限单元法的求解过程结束。

现在给出第一类边界条件、三角剖分、线性插值的位场延拓的有限单元法程序框图如下：



## 2.7 计算实例

对于下图所示区域，用三角单元进行剖分，剖分后的节点分布及单元编号如该图所示。区域边界上的节点的场值是已知的（本例按某磁性体的理论公式算出）。现计算区域内部节点的场值。



首先，列出参数如下：

（1）节点总数；

（2）单元总数；

（3）单元节点编号[数组]；

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **单元序号** | 1 | 2 | 3 | 4 | … | … | 26 | 27 | 28 |
|  | 1 | 1 | 7 | 7 | … | … | 18 | 18 | 18 |
|  | 4 | 5 | 5 | 5 | … | … | 19 | 14 | 17 |
|  | 5 | 2 | 4 | 2 | … | … | 17 | 16 | 16 |

(4) 节点坐标[数组]；

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **节点序号** | 1 | 2 | 3 | 4 | … | … | 19 | 20 | 21 |
|  | 0 | 0 | 0 | 1 | … | … | 4 | 3.5 | 4 |
|  | 0 | 1 | 2 | 0.5 | … | … | 1.2 | 0.8 | 0.5 |

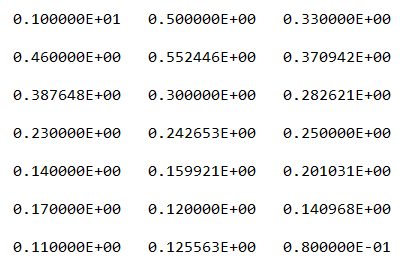
（5）第一类边界节点数；

（6）第一类边界节点号[数组]和场值[数组]；

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **第一类边界**  **节点序号** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| **第一类边界**  **节点号** | 1 | 2 | 3 | 4 | 8 | 10 | 12 | 13 | 16 | 17 | 19 | 21 |
| **场值** | 1 | 0.5 | 0.33 | 0.46 | 0.3 | 0.25 | 0.23 | 0.14 | 0.17 | 0.08 | 0.11 | 0.12 |

（7）组合成整体计算程序。

计算结果如下



利用剖分和编号程序对区域进行更加复杂的剖分，并赋予第一类边界条件 ，其中ND = 20201，NE = 40000，ND1 = 400，重新计算并绘制节点处场值的离散点图如下：

图片包含 屏幕截图, 电子产品

已生成高可信度的说明

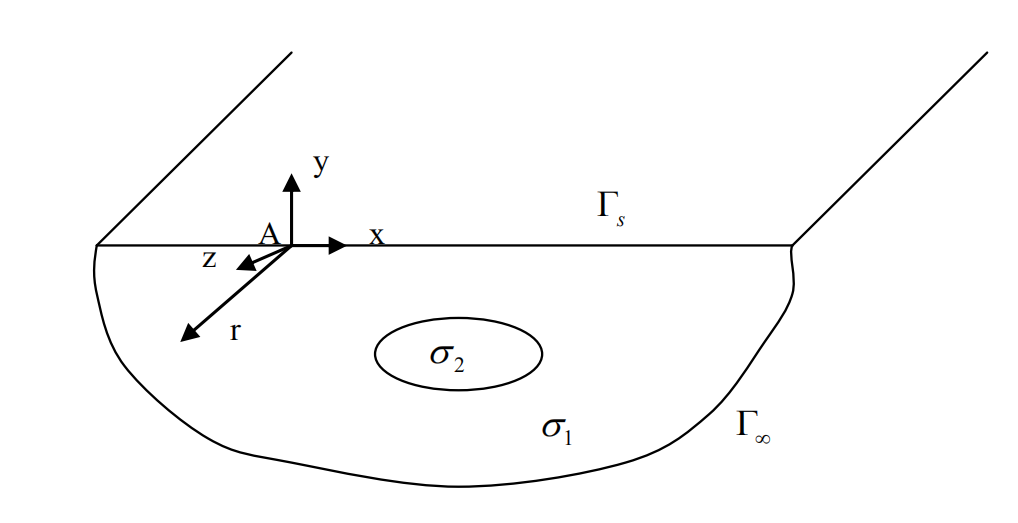
# 第三章 电阻率模拟

## 3.1 二维均匀电场

中间梯度法是电阻率法中最常用的方法之一，特别适于寻找直立的高阻脉和水平的良导层。当MN的测量范围在AB/3的中间区域，电位近似于线性变化。所以，初始电场可以近似看作水平均匀电场，是最简单的电场。本节将从最简单的电场开始介绍有限单元法在电阻率模拟中的应用，并用matlab实现编程。

**3.1.1 问题**

设在地面 A 点，置一电流强度为 I 的点电源，地下构造成二维分布，如图所示：



**图3.1二维模型**

构造泛函：



对上面的问题求变分，可得：



根据已知条件，代入可得：

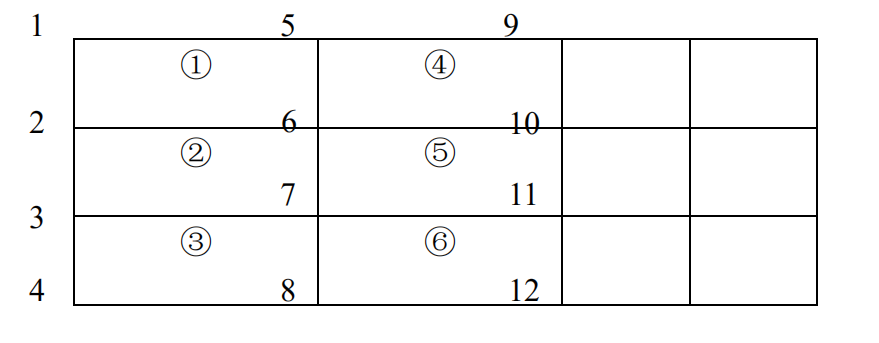


上式子表面，内部边界在变分中不再出现。所以只与外部边界有关，这是有限单元法能够处理物性复杂分布区域的依据。将边界条件代入，可得变分问题：



**3.1.2 模型的建立**

本文主要从双线性的矩形单元对区域进行剖分，每个单元的四个角点为节点，在单元内进行双线性插值，如图：



**图3.2剖分示意图**

其中采用单元的编号和整体编号与上图的方式一样，但赋值的过程本文采用先赋背景值，再将大异常赋值，再赋其他小异常的值，这样可以减少matlab的循环次数，从而进行速度的提升。

代码：

function [I,I\_point,I\_xy]=Poufen(x,y)

n\_x=length(x);

n\_y=length(y);

I=(n\_x-1)\*(n\_y-1);

I\_point=zeros(4,I);

I\_xy=zeros(2,n\_x\*n\_y);

for i=1:1:n\_x-1

I\_point(1,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (1+(i-1)\*n\_y:i\*n\_y-1); %I\_point(m,n),m代表节点编号，n代表单元号

I\_point(2,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (2+(i-1)\*n\_y:i\*n\_y); %m是4，n等于单元总数,这步时给每个单元的节点编号

I\_point(4,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (1+i\*n\_y:(i+1)\*n\_y-1);

I\_point(3,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (2+i\*n\_y:(i+1)\*n\_y);

end

% 给每个编号节点赋值坐标I\_xy(m,n)，m是2，代表维数，n代表节点编号

for i=1:1:n\_x

I\_xy(1,(i-1)\*n\_y+1:i\*n\_y) = x(i);

I\_xy(2,(i-1)\*n\_y+1:i\*n\_y) = y(1:n\_y);

end

End

**3.1.3 模型计算**

均匀半空间

Earth

Air



E=1的均匀电场

**图3.2均匀半空的模型和视电阻率图**

可以看到，当为均匀半空间时，其电导率全为1，符合真实的情况，所以，这也一方面验证了有限单元法的有效性。

0.1

Earth

Air



E=1的均匀电场



**图3.3高阻异常的模型和视电阻率图**

10

Earth

Air



E=1的均匀电场



**图3.4低阻异常的模型和视电阻率图**

可以看到，当为高阻的时候，视电阻率是不能很好的反映出异常体的真实电阻率，而当为低阻时却能比较接近。此外，从视电阻率曲线来看，在异常的两端，视电阻率都会产生一个小的回跳。从异常的位置和视电阻率图来看，能够非常明显地反映出异常体的位置，这也说明了有限单元法在中间梯度法中的应用是可行和合理的。

## 3.2 点源二维电场

有一定走向的构造称为二维构造，可以用二个坐标变量描述构造地电特性。但是电阻率法中常用点电源供电，点电源产生的电场是三维的，所以，点源二维构造的电场实质上是三维的，它有三个坐标变量。用有限单元法解决三维问题是比较麻烦的，需要庞大的计算机内存。但是对于点源二维构造的电场，可用傅里叶变换的方法，将三维电场变换成带参数的二维问题，便于用有限单元法求解。

**3.2.1 问题**

设在地面 A 点，置一电流强度为 I 的点电源，地下构造成二维分布，如图3.1所示。

将三维问题经过傅里叶变换后，三维边值问题变为下列二维边值问题：



在未进行傅里叶变换的三维问题中，其中和是三维区域的边界，A是点电源的位置，此外上式的区域是二维的，所以边界和是二维区域的边界。

用有限单元法求解上面的边值问题，首先要将它变为变分问题，首先，构造一个泛函：





式子中的n代表剖分区域的外法向。将边值问题中的外部边界条件代入上式中，可得：



所以有：



所以，可得到变分问题为：





**3.2.2 傅里叶反变换**

利用有限单元法可以解决上面的变分问题，可获得各节点的U，该U对应于特定的波数k。如果计算了对应于不同k的一组U，可用傅里叶反变换计算三维空间的电位u。

用数值方法进行傅里叶反变换，波数k的选择是保证计算精度和节省计算时间的主要问题。现介绍最优化法选择k的思路（徐世浙，1988）。

通常，我们对通过电源点的剖面上的电位感兴趣。这时z=0.所以由傅里叶反变换式：



在数值方法中，上式的积分使用数值积分完成的。将上式右侧写成近似式：



其中，是离散的值，是系数。我们要选择适当的和，使上式在r的一定范围内尽可能准确。但是，在一般情况下，函数u和U的形式使未知的，所以无法利用上面式子来选择和。

在均匀半空间情形下，



代入傅里叶变换中，可得：



其中是主剖面上的点至电源点的距离，将上面式子代入之中，可得：



将上式写成数值积分的近似式：



用最优化方法选择和，使上式在r的一定范围内尽可能准确。由此得出的和，作为一般情形下，傅里叶反变换式子中的和。

为了在不同的r下，有相同的相对误差，将上式子写成：



选取一系列，得方程组：



其中，,,,，，选取和，使目标函数：



取极小，其中是单位列向量。

和得选择分两步进行。第一步，给定一组，于是为已知，由的极小，决定一组，此时，用的是线性最小二乘法。第二步，研究在什么样的一组下，目标函数取极小，此时，用的是非线性最小二乘法。如此反复迭代若干次，可获得最佳的一组和，用这种方法，决定的一组和，如表3.1所示，这组和使r在范围内，傅里叶反变换的相对误差在0.1%以内。

**表3.1用最优化方法选择的一组和**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| k | 0.004758 | 0.0407011 | 0.1408855 | 0.393225 | 1.0880380 |
| g | 0.0099472 | 0.0381619 | 0.0980327 | 0.2511531 | 0.7260814 |

**3.2.3 模型计算**

均匀半空间

Earth

Air



**图3.5均匀半空的模型和响应图**

D

Air





**图3.6 D模型和响应图**

G

Air





**图3.7 G模型和响应图**

低阻异常

Earth

Air





**图3.8低阻异常的模型和响应图**

高阻异常体

Earth

Air





**图3.9高阻异常的模型和响应图**

可以分析得到，但为均匀半空间时，由于中间的点为点电源的位置，所以会上升得非常快，有尖锐，这是奇异性所造成的，而U从电源向四周在逐渐减小，这是符合真实情况的。当为2层模型时，可以看到，但第二层为高阻时，会阻碍电流向下流动，故电流主要会集中到第一层，所以响应大小比均匀半空间时大；而第二层为低阻时，会吸引电流流向第二层，所以响应大小比均匀半空间时小。当模型是由低阻异常时，可以看到场值更加集中，曲线变得更加尖锐，但响应变小，这是由于低阻将吸引了一部分电流；而模型为高阻异常时，可以看到曲线尖锐部分变宽，并且响应变大，这是因为高阻体的‘排斥’效果。

# 第四章 直接时间域矢量有限元瞬变电磁三维正演

## 4.1磁场的双旋度方程

在三维各向同性介质中，时谐变因子为时，电流源的电磁场满足的Maxwell方程组为



式中，为介质的介电常数，为电导率，为磁导率，为激励电性源的圆频率，为虚数单位，为时间，和分别为电场和磁场矢量，为外加电流源的电流密度。一般利用节点有限元法求解（1）式时，为了规避计算电流密度的旋度或电场法向不连续的问题，常先计算电场或矢量势、然后通过求导来获取磁场值。然而，由于求导运算，计算得出的磁场值将比直接用有限元求解的量精度低。本文拟直接利用有限元法计算磁场值，并引入伪源解决电流密度的旋度计算问题。由于不存在磁荷和面电流密度，磁场在在内部电性界面上，同时具有法向和切向连续性。为通用起见，在外边界上给出磁场的第三类边界条件为



其中为的外法向单位向量，为关于边界条件的已知量。利用代入法消去式（1）中电场项，即可得到磁场满足的双旋度方程



其中，。

## 4.2 有限单元法

### 4.2.1 变分原理

根据变分原理，通过求解泛函（4）的驻点，能够获得满足方程（3）和边界条件（3）的磁场解



其中，表示电流密度的旋度。在有源问题中，为简单起见，可以将计算区域剖分到较远的范围，然后略去边界积分，于是泛函（4）可简化为



### 4.2.2 网格剖分

本文中为程序简便起见，采用规则六面体单元剖分整个区域。为了在保证计算精度的基础上尽可能地减小计算量，我们把整个研究区域分为两部分：一部分是目标区域，一部分是网格边界区域。如图1所示，在目标区域，我们要求能获得近似准确的磁场值，因而采用较为细致、均匀的网格剖分；网格边界区域是为了提高数值模拟精度、减小边界影响而向外延伸的区域，我们将网格向着外边界迅速变得稀疏，使外边界尽量远离目标区域，以减少变分（4）中忽略边界积分的影响。

图1 网格剖分示意图



扩展区域

目标区域

### 4.2.3 单元分析

在每个小单元内有线性方程组

 (6)

其中，为单元内号节点的形函数，为单元内节点个数，分别计算各单元子矩阵并合成总体方程，求解即可获得各节点处的磁场值。

### 4.2.4 总体合成

对剖分后的单元进行总体合成，记









则任意单元的离散化方程（7），可最终写成

 (7)

总体合成就是将方程（7）中的各元素按节点的网格排列顺序放到求解域的整体系数矩阵中，与对应的节点元素叠加，最终的矩阵方程包括所有单元内所有节点的场方程。

## 4.3 回线源的加载

电磁勘探中一般使用体积可忽略的细导线供电，这给源的加载带来了困难。本文采用一个近似函数来等效场源的作用，将源分布于一定范围内，避免在源点处产生奇异性，从而不用区分背景场和异常场，直接求解总场值[7]，伪函数的表达式为

 (8)

参数是控制源分布宽度和幅值的参数，为某一空间维度的坐标，为源所在位置即奇异点的坐标值。

因此，空间处电流强度为、沿着方向的导线源的响应，其电流密度的分布可表表为。那么对于回线源而言，其实就是四根沿着一致方向（顺时针或逆时针）的导线源的叠加。

显然，使用伪函数描述源项，相当于将导线进行了一定程度的展宽。在离场源较远的位置，这种展宽的效应可以忽略。而式（7）中对电流密度求旋度的运算，可转换为对伪函数的导数运算

 (9)

## 4.4 散度条件的施加

在推导式（6）时，我们取泛函（5）的一阶变分，这意味着试探场必须二阶可微。但实际上在有限单元法的数值模拟中，离散化时只要求插值函数或展开函数连续，而对其导数未作任何要求，这样得出的解是不严格满足控制微分方程的“弱解”。“弱解”有时是错误的，不满足散度条件。

由广义变分原理建立泛函



按照第二部分同样的过程，可以将其转化为对应的刚度矩阵

 (10)

将方程（10）加入到方程（7）中，即完成了散度条件的施加。

# 第五章 面向目标的自适应网格剖分

## 5.1 前沿

5.1.1 选题背景及研究意义

近年来，三维正演得到了快速的发展，这即得益于计算机的发展，而得益于有限单元法、有限体积法等理论在地球物理学的发展。随着计算精度和复杂模型复杂度的提高，近年来非结构网格剖分得到良好的发展。

为什么要进行自适应网格剖分？

主要原因分为以下几点：

1.三维正演通常需要较大的内存，为了节省计算资源，应合理的利用较少的单元来得到较好的结果，所以自适应网格剖分今年来得到较大的发展。

2.3维反演时，由于无法确定异常的位置，若不自适应，则每进行一次正演都需要进行一次手工网格剖分，大大的降低效率和人力。

一般自适应网格剖分的局限性：

一般使用的时全局自适应网格剖分，据经验和一些理论证明，当网格剖分细化到一定程度之后，精确度主要受局部网格的剖分有较大的关系，而与其他地方的剖分无关，这就导致了往后的对网格的细化有较大的浪费。所以这就引出了面向目标的自适应网格剖分，上面的思想也就是面向目标的自适应网格剖分的核心思想。

5.1.2 误差分析

1.网格剖分：

（i）剖分边界距离异常体不够远（即剖分规模不够大，一般要距离5到10个趋肤深度远）；

（ii）剖分单元的大小比一个趋肤深度的大小大（剖分单元应该要小于一个趋肤深度，这样才能保证电场的近似线性变化）；

（iii）每个剖分单元的质量（要符合物理定律，例如，三角形单元同一个单元内，3个电场点环绕圆的中心是磁场，所以圆心应该在该单元里面）；

√

×

**图5.1 剖分单元示意图**

（iv）对曲面的剖分不够好，未能准确地反映地形的起伏变化。

2.最终的方程组的求解不准确。（可以利用MKL的数学库中多核并行的LU求解器PARDISO来进行求解，可以得到更高精度的解）

## 5.2 后验误差估计的理论

5.2.1 控制网格细化的变量是一种后验误差值

按照Maxwell方程组来说，从一个单元一个面上到另一个单元该面的另一边上，其电流密度垂向分量和磁场强度的切向分零六是连续的，即：



但实际单元剖分时，是未满足该条件的，所以基于这2个条件提出了3种类型的基于残差的后验误差估计方式：

（1）测量总电流密度的法向分量和磁场强度的切向分量的连续（rJH方法）；



（2）测量总电流密度的无发散条件在多大程度上得到满足（J方法）；



（3）测量对磁场基本连续性条件（H方法）；



5.2.2 面向目标的后验误差估计

构造一种评价所在单元内电场大小的线性泛函L(E)，则自适应算法的目标是为减少误差函数L(e)，其中表示正演模拟解与精确解之间的差距，并利用对偶关系，可得：









其中，就是基于面向目标的自适应后验误差估计算子，而可以根据前面的后验误差估计种类来得到，而则是一种加权系数，在物理上，则等效于在这些子域的范围内引入子源。

5.2.3 自动网格加密的方法分类

1.全局网格加密方式：没有后验误差估计，当全局误差当未达到阈值，则所有的单元只是简单的一分为二。

2.非面向目标的rJH方法：令，只用rJH的后验误差估算；

3.非面向目标的J方法：令，只用J的后验误差估算；

4.非面向目标的H方法：令，只用H的后验误差估算；

5.面向目标的rJH方法：用rJH的后验误差计算、，网格细化由来决定；

6.面向目标的J方法：用J的后验误差计算、，网格细化由来决定；

7.面向目标的H方法：用H的后验误差计算、，网格细化由来决定；

5.2.4 面向目标的自动网格加密的流程

其中，全局相对误差估计的计算公式为：



其中，每个单元的误差指示器为：





**图9.2 面向目标的自适应网格剖分流程图**

# 第六章 任意发射波形的视电阻率定义

## 6.1 视电阻率定义理论

于中心回线装置的视电阻率定义，由于其分量分量关于电阻率是单调函数，所以根据文献[14]，可将关于视电阻率变量的场值按照泰勒公式张开，可得：



其中，，是自己给的初始模型的电阻率值，是正演的场值，是真实的视电阻率，则是观测值。对于上面的式子，我们只取其中的线性部分，所以式子变为：



对于该式子，一开始，是我们给与的初始模型的电阻率，是已知的观测值，并且对于其中的一阶导数，我们可以使用向后差分代替，所以可以得到迭代公式：





其中是代表第次迭代时的电阻率，是第次迭代时的由式子()计算得到的。其中，迭代的终止条件为：



是迭代终止误差。

对于迭代过程中未达到精度要求的值，为了得到完整的视电阻率曲线，可以利用最小曲率插值方法（多辐射场源地空瞬变电磁法多分量全域视电阻率定义）进行补齐空值，公式如下：





其中k是迭代次数，M是总计算点数。

## 6.2 三维计算模型

本文中利用有限单元法进行各个波形三维的正演，并将各个波形的三维数据进行视电阻率定义，其中模型的设置为：假设在均匀大地的下方存在一个的长方形的异常体，其埋深为，异常体的电阻率为，均匀大地的电阻率，使用中心回线装置在30m的高空进行测量。示意图如下：

图片包含 屏幕截图

描述已自动生成

**图6.1.三维模型示意图**

下面给出了各种发射波形的视电阻率定义图其视电阻率图：

图片包含 屏幕截图

描述已自动生成

**图6.2.阶跃波视电阻率图**

图片包含 屏幕截图

描述已自动生成

**图6.3.半正弦波视电阻率图**

图片包含 屏幕截图

描述已自动生成

**图6.4.三角波视电阻率图**



**图6.5.梯形波视电阻率图**

图片包含 屏幕截图

描述已自动生成

**图6.6.伪随机波波视电阻率图**

从上面几个图可以分析得到，（1）各种波形的视电阻率图都能看到明显的异常；（2）不同的波形，其视电阻率效果不同，其中三角波和半正弦波视电阻率图接近，但效果差，阶跃波和伪随机波的视电阻率图比较接近，效果好，而梯形波介于前面2类之间；（3）各种波形都能较好的定位到异常的地表位置，但同时也能看到半正弦波、三角波和梯形波都不能较好的反映异常的深度，会导致异常的深度稍微往上偏移，并且将异常的影响范围给拉伸了。其中，半正弦波和三角波比较接近，而梯形波被拉伸的范围要小于它们，而伪随机波不存在这种情况。伪随机波的视电阻率效果图与阶跃波很接近，不仅能够很好地反映异常的地表位置，还能很好地反映出异常的深度。所以，伪随机波不仅能模拟真实波形，抗噪强等优点，而且其视电阻率图接近阶跃波的视电阻率图，利用分析。

# 参考文献

[1]李貅.瞬变电磁测深的理论与应用[M].西安:陕西科学技术出版社,2002.

[2]徐世浙. 地球物理中的有限单元法[M]. 北京: 科学出版社, 1994.

[3]朴化荣. 电磁测深法原理[M]. 北京: 地质出版社, 1990.

[4]朱伯芳. 有限单元法原理与应用[M]. 北京: 水利出版社, 1979.

[5]李金铭. 地电场与电法勘探[M]. 北京: 地质出版社, 2005.

[6]徐世浙. 地球物理中的有限单元法[M]. 北京: 科学出版社, 1994.

[7]徐士良. FORTRAN 常用算法程序集[M]. 清华大学出版社, 1995.

[8]姚伟华 瞬变电磁法矢量有限元三维正演研究[D]. 西安: 长安大学, 2015.

[9]Ren Z, Kalscheuer T, Greenhalgh S and Maurer H 2013 A goal-oriented adaptive finite-element approach for plane wave 3-d electromagnetic modeling Geophys. J. Int. 194 700–18.

[10]Kerry Key and Jeffrey Ovall, A parallel goal-oriented adaptive finite element method for 2.5-D electromagnetic modelling, Geophys. J. Int., 186(1),137–154.

[11]Key, K., 2016, MARE2DEM: a 2-D inversion code for controlled-source electromagnetic and magnetotelluric data: Geophysical Journal International, 207(1)，ggw290.

[12]张莹莹,李貅,姚伟华,智庆全,李佳.多辐射场源地空瞬变电磁法多分量全域视电阻率定义[J].地球物理学报,2015,58(08):2745-2758.

[13]齐彦福. 复杂介质中时间域航空电磁数据仿真技术研究[D].吉林大学,2017.

# 附录

**二维中间梯度电场有限元Fortran代码：**

clc;

clear;

close all;

I0=1; %电流大小

x=(0:50); %x的坐标

y=(0:-1:-10); %y的坐标

nx=length(x); %x方向的点数

ny=length(y); %y方向的点数

n\_point=nx\*ny; %总共节点点数

I= (nx-1)\*(ny-1); %单元数

E0=1;

[I,I\_point,I\_xy]=Poufen(x,y);

cond1=1;

cond2=0.1;

I\_normal=[232:234,242:244,252:254,262:264,272:274];

I\_cond=Fuyu(cond1,cond2,I\_normal,I);

a=zeros(1,I);

b=zeros(1,I);

for i=1:1:I

a(i)=I\_xy(1,I\_point(4,I))-I\_xy(1,I\_point(1,I));

b(i)=I\_xy(2,I\_point(1,I))-I\_xy(2,I\_point(2,I));

end

n\_x=length(x);

n\_y=length(y);

I\_bound\_left=(1:n\_y-1); %给出左边界的单元,根据需要进行修改

I\_bound\_right=(I-n\_y+2:I); %给出右边界的单元，根据需要进行修改

K=Inside(I,I\_cond,a,b); %构建每个单元的K矩阵

P=Bound(I,I\_cond,a,b,I\_bound\_left,I\_bound\_right,E0); %构建每个单元的非边界系数矩阵P

[Ke,Pe]=Unit\_syn(K,P,I,I\_point); %单元合成

u=Ke\Pe;

U\_P=u(1:ny:n\_point);

K\_mn=-1.0;

[ps,x\_u]=trsps(U\_P,K\_mn,x,I0);

plot(x\_u,-1.0./ps)

title('中间梯度法');

xlabel('x')

ylabel('电导率');

function P=Bound(I,I\_cond,a,b,I\_left,I\_right,E0)

P=zeros(4,I); %第二个下标代表第i个单元的P向量

% 给出边界start

P(1,I\_left(1:end))=1;

P(2,I\_left(1:end))=1;

P(3,I\_right(1:end))=-1;

P(4,I\_right(1:end))=-1;

% 给出边界end

for i=1:1:I

P(:,i)=P(:,i)\*I\_cond(i)\*b(i)/2;

end

end

% 此子程序是给每一个单元赋予电导率

% 采用的是覆盖方法，从大到小一层层覆盖赋值

function I\_cond=Fuyu(cond1,cond2,I\_normal,I)

I\_cond=zeros(1,I);

I\_cond(1:end)=cond1;

n\_normal=length(I\_normal);

for k=1:1:n\_normal

I\_cond(1,I\_normal(k))=cond2;

end

End

function K=Inside(I,I\_cond,a,b)

K=zeros(4,4,I);

alph=I\_cond.\*b/6./a;

beta=I\_cond.\*a/6./b;

K(1,1,:)=2\*alph+2\*beta;

K(2,1,:)=alph-2\*beta;

K(3,1,:)=-alph-beta;

K(4,1,:)=-2\*alph+beta;

K(2,2,:)=K(1,1,:);

K(3,2,:)=K(4,1,:);

K(4,2,:)=K(3,1,:);

K(3,3,:)=K(1,1,:);

K(4,3,:)=K(2,1,:);

K(4,4,:)=K(1,1,:);

for i=1:1:I

K(:,:,i)=K(:,:,i)+triu(K(:,:,i).',1); %构建对称矩阵

end

End

function [I,I\_point,I\_xy]=Poufen(x,y)

n\_x=length(x);

n\_y=length(y);

I=(n\_x-1)\*(n\_y-1);

I\_point=zeros(4,I);

I\_xy=zeros(2,n\_x\*n\_y);

for i=1:1:n\_x-1

I\_point(1,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (1+(i-1)\*n\_y:i\*n\_y-1); %I\_point(m,n),m代表节点编号，n代表单元号

I\_point(2,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (2+(i-1)\*n\_y:i\*n\_y); %m是4，n等于单元总数,这步时给每个单元的节点编号

I\_point(4,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (1+i\*n\_y:(i+1)\*n\_y-1);

I\_point(3,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (2+i\*n\_y:(i+1)\*n\_y);

end

% 给每个编号节点赋值坐标I\_xy(m,n)，m是2，代表维数，n代表节点编号

for i=1:1:n\_x

I\_xy(1,(i-1)\*n\_y+1:i\*n\_y) = x(i);

I\_xy(2,(i-1)\*n\_y+1:i\*n\_y) = y(1:n\_y);

end

End

function [ps,x\_u]=trsps(U\_P,K\_mn,x,I0)

n\_x=length(x);

x\_u=zeros(1,n\_x-1);

for i=1:1:n\_x-1

x\_u(i)= (x(i+1)+x(i))/2.d0;

ps(i)=( U\_P(i+1)-U\_P(i) )/( x(i+1)-x(i) );

end

end

% 合成的思路：

% 将每个单元的矩阵和向量先拆解为方程组，再将所有的方程组组合起来

% 最后进行单元的合成就能变为一个大的矩阵

% 这里的细节是：将边界处耦合起来，其实就是将边界处不同单元的几个方程加起来变为一个

function [Ke,Pe]=Unit\_syn(K,P,I,I\_point)

n\_point=I\_point(3,end); %提取总节点数

Pe=zeros(n\_point,1);

Ke=zeros(n\_point,n\_point);

for i=1:1:I

I1=I\_point(1,i);

I2=I\_point(2,i);

I3=I\_point(3,i);

I4=I\_point(4,i);

Ke(I1,[I1,I2,I3,I4])=Ke(I1,[I1,I2,I3,I4])+K(1,:,i);

Ke(I2,[I1,I2,I3,I4])=Ke(I2,[I1,I2,I3,I4])+K(2,:,i);

Ke(I3,[I1,I2,I3,I4])=Ke(I3,[I1,I2,I3,I4])+K(3,:,i);

Ke(I4,[I1,I2,I3,I4])=Ke(I4,[I1,I2,I3,I4])+K(4,:,i);

Pe([I1,I2,I3,I4])=Pe([I1,I2,I3,I4])+P(:,i);

end

end

**二维电源电场有限元Fortran代码：**

clc;

clear;

close all;

k=[0.004758 0.0407011 0.1408855 0.393225 1.0880380];

g=[0.0099472 0.0381619 0.0980327 0.2511531 0.7260814];

I0=1; %电流大小

x=(0:200)\*0.1; %x的坐标

y=(0:-1:-10)\*0.1; %y的坐标

nx=length(x); %x方向的点数

ny=length(y); %y方向的点数

n\_point=nx\*ny; %总共节点点数

I= (nx-1)\*(ny-1); %单元数

n\_source=(ceil((ny)/2)-1)\*(nx-1)+1;

source=[n\_source]; %给定源所在的单元，并以第1节点作为源的位置；

E0=1;

[I\_point,I\_xy]=Poufen(x,y); %只适用于矩形的线性剖分

cond=[1 0.05];

n\_cond=length(cond);

I\_normal=[3+10\*(0:nx-2) 4+10\*(0:nx-2) 5+10\*(0:nx-2) 6+10\*(0:nx-2) 7+10\*(0:nx-2) 8+10\*(0:nx-2) 9+10\*(0:nx-2) 10+10\*(0:nx-2)]; %I\_normal有n行，即代表有n个异常

I\_cond=Fuyu(cond,I\_normal,I); %I\_cond存储着每个单元的电导率

a=zeros(1,I);

b=zeros(1,I);

for i=1:1:I

a(i)=I\_xy(1,I\_point(4,I))-I\_xy(1,I\_point(1,I)); %计算每个单元的边长

b(i)=I\_xy(2,I\_point(1,I))-I\_xy(2,I\_point(2,I));

end

n\_x=length(x);

n\_y=length(y);

I\_bound\_left=(1:n\_y-1); %给出左边界的单元,根据需要进行修改

I\_bound\_right=(I-n\_y+2:I); %给出右边界的单元，根据需要进行修改

I\_bound\_bottom=(ny-1)\*(1:nx-1); %给出底边界的单元，根据需要进行修改

K1=Inside1(I,I\_cond,a,b); %构建每个单元的K矩阵

Ue=0;

for i=1:1:length(k)

Ke=0;

P=zeros(n\_point,1);

K2=Inside2(I,I\_cond,a,b,k(i)); %构建每个单元的K矩阵

K3=Inside3(I,I\_cond,k(i),source,I\_point,I\_xy,I\_bound\_left,I\_bound\_right,I\_bound\_bottom);

P(I\_point(1,source))=0.5;

Ke=Unit\_syn(K1,K2,K3,I,I\_point); %单元合成

Ue=Ue+Ke\P\*g(i);

end

u=Ue(1:ny:n\_point);

K\_mn=1.0;

[ps,x\_u]=trsps(u,K\_mn,x,I0);

plot(x,u)

function [I\_point,I\_xy]=Poufen(x,y)

n\_x=length(x);

n\_y=length(y);

I=(n\_x-1)\*(n\_y-1);

I\_point=zeros(4,I);

I\_xy=zeros(2,n\_x\*n\_y);

for i=1:1:n\_x-1

I\_point(1,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (1+(i-1)\*n\_y:i\*n\_y-1); %I\_point(m,n),m代表节点编号，n代表单元号

I\_point(2,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (2+(i-1)\*n\_y:i\*n\_y); %m是4，n等于单元总数,这步时给每个单元的节点编号

I\_point(4,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (1+i\*n\_y:(i+1)\*n\_y-1);

I\_point(3,(i-1)\*(n\_y-1)+1:i\*(n\_y-1)) = (2+i\*n\_y:(i+1)\*n\_y);

end

% 给每个编号节点赋值坐标I\_xy(m,n)，m是2，代表维数，n代表节点编号

for i=1:1:n\_x

I\_xy(1,(i-1)\*n\_y+1:i\*n\_y) = x(i);

I\_xy(2,(i-1)\*n\_y+1:i\*n\_y) = y(1:n\_y);

end

End

% 此子程序是给每一个单元赋予电导率

% 采用的是覆盖方法，从大到小一层层覆盖赋值

function I\_cond=Fuyu(cond,I\_normal,I)

I\_cond=zeros(1,I);

I\_cond(1:end)=cond(1);

n\_cond=length(cond);

for i=2:1:n\_cond

n\_normal=length(I\_normal(i-1,:));

for k=1:1:n\_normal

I\_cond(1,I\_normal(k))=cond(i);

end

end

End

function K=Inside1(I,I\_cond,a,b)

K=zeros(4,4,I);

alph=I\_cond.\*b/6./a;

beta=I\_cond.\*a/6./b;

K(1,1,:)=2\*alph+2\*beta;

K(2,1,:)=alph-2\*beta;

K(3,1,:)=-alph-beta;

K(4,1,:)=-2\*alph+beta;

K(2,2,:)=K(1,1,:);

K(3,2,:)=K(4,1,:);

K(4,2,:)=K(3,1,:);

K(3,3,:)=K(1,1,:);

K(4,3,:)=K(2,1,:);

K(4,4,:)=K(1,1,:);

for i=1:1:I

K(:,:,i)=K(:,:,i)+triu(K(:,:,i).',1); %构建对称矩阵

end

End

function K=Inside2(I,I\_cond,a,b,k)

K=zeros(4,4,I);

alph=a.\*b.\*k.\*k.\*I\_cond/36;

K(1,1,:)=4\*alph;

K(2,1,:)=alph\*2;

K(3,1,:)=alph;

K(4,1,:)=2\*alph;

K(2,2,:)=K(1,1,:);

K(3,2,:)=K(4,1,:);

K(4,2,:)=K(3,1,:);

K(3,3,:)=K(1,1,:);

K(4,3,:)=K(2,1,:);

K(4,4,:)=K(1,1,:);

for i=1:1:I

K(:,:,i)=K(:,:,i)+triu(K(:,:,i).',1); %构建对称矩阵

end

End

function K=Inside3(I,I\_cond,k,source,I\_point,I\_xy,I\_left,I\_right,I\_bottom)

K=zeros(4,4,I);

xy\_source=I\_xy(:,I\_point(1,source));

xy\_left=I\_xy(:,I\_point(1,I\_left));

xy\_right=I\_xy(:,I\_point(1,I\_right));

xy\_bottom=I\_xy(:,I\_point(1,I\_bottom));

n\_left=length(I\_left);

n\_right=length(I\_right);

n\_bottom=length(I\_bottom);

% 左边界处理 start

for i=1:1:n\_left

I=I\_left(i);

r=sqrt( (xy\_left(1,i)-xy\_source(1))^2+(xy\_left(2,i)-xy\_source(2))^2 ); %求取点距

cosa=abs(xy\_left(1,i)-xy\_source(1))/r; %求取夹角cos

b=I\_xy(2,I\_point(1,I))-I\_xy(2,I\_point(2,I));

beta=b\*I\_cond(I)\*k\*bessely(1,k\*r)\*cosa/6/bessely(0,k\*r); %bessely第二类bessel函数

K(1,1,I)=2\*beta;

K(2,1,I)=beta;

K(2,2,I)=K(1,1,I);

end

% 左边界处理 end

% 右边界处理 start

for i=1:1:n\_right

I=I\_right(i);

r=sqrt( (xy\_right(1,i)-xy\_source(1))^2+(xy\_right(2,i)-xy\_source(2))^2 ); %求取点距

cosa=abs(xy\_right(1,i)-xy\_source(1))/r; %求取夹角cos

b=I\_xy(2,I\_point(1,I))-I\_xy(2,I\_point(2,I));

beta=b\*I\_cond(I)\*k\*bessely(1,k\*r)\*cosa/6/bessely(0,k\*r); %bessely第二类bessel函数

K(1,1,I)=2\*beta;

K(2,1,I)=beta;

K(2,2,I)=K(1,1,I);

end

% 右边界处理 end

% 底边界处理 start

for i=1:1:n\_bottom

I=I\_bottom(i);

r=sqrt( (xy\_bottom(1,i)-xy\_source(1))^2+(xy\_bottom(2,i)-xy\_source(2))^2 ); %求取点距

sina=abs(xy\_bottom(2,i)-xy\_source(2))/r; %求取夹角cos

a=I\_xy(1,I\_point(4,I))-I\_xy(1,I\_point(1,I));

beta=a\*I\_cond(I)\*k\*bessely(1,k\*r)\*sina/6/bessely(0,k\*r); %bessely第二类bessel函数

K(1,1,I)=2\*beta;

K(2,1,I)=beta;

K(2,2,I)=K(1,1,I);

end

% 底边界处理 end

for i=1:1:I

K(:,:,i)=K(:,:,i)+triu(K(:,:,i).',1); %构建对称矩阵

end

End

% 合成的思路：

% 将每个单元的矩阵和向量先拆解为方程组，再将所有的方程组组合起来

% 最后进行单元的合成就能变为一个大的矩阵

% 这里的细节是：将边界处耦合起来，其实就是将边界处不同单元的几个方程加起来变为一个

function Ke=Unit\_syn(K1,K2,K3,I,I\_point)

n\_point=I\_point(3,end); %提取总节点数

Ke=zeros(n\_point,n\_point);

for i=1:1:I

I1=I\_point(1,i);

I2=I\_point(2,i);

I3=I\_point(3,i);

I4=I\_point(4,i);

Ke(I1,[I1,I2,I3,I4])=Ke(I1,[I1,I2,I3,I4])+K1(1,:,i)+K2(1,:,i)+K3(1,:,i);

Ke(I2,[I1,I2,I3,I4])=Ke(I2,[I1,I2,I3,I4])+K1(2,:,i)+K2(2,:,i)+K3(2,:,i);

Ke(I3,[I1,I2,I3,I4])=Ke(I3,[I1,I2,I3,I4])+K1(3,:,i)+K2(3,:,i)+K3(3,:,i);

Ke(I4,[I1,I2,I3,I4])=Ke(I4,[I1,I2,I3,I4])+K1(4,:,i)+K2(4,:,i)+K3(4,:,i);

end

End

function [ps,x\_u]=trsps(U\_P,K\_mn,x,I0)

n\_x=length(x);

x\_u=zeros(1,n\_x-1);

for i=1:1:n\_x-1

x\_u(i)= (x(i+1)+x(i))/2.d0;

ps(i)=( U\_P(i+1)-U\_P(i) )/( x(i+1)-x(i) );

end

end