

在IBM-PC微机上进行 二维电法有限元正演计算的讨论

中国矿业大学 韩恩杰

近年来,二维电法有限元正演模拟技术发展较快。但是,由于该问题数据量庞大、占机器内存多,其计算基本上都是在大型计算机或小型计算机上进行。就我国目前情况而言,大多数生产单位还不具备以上条件,而微型计算机已十分普及。因此,如何在微型计算机上开展二维电法研究工作,已成为该方法推广应用中所亟待解决的问题。

众所周知,在IBM-PC微机上使用磁盘BASIC或高级BASIC时,内存最大工作空间为64KB,扣除4KB用作BASIC解释程序工作区,则只有60KB内存可供用户程序使用。事实上,对于实型数,即使60KB内存全部用于存放数据也只能容纳 $(60 \times 1024) / 4 = 15360$ 个数据,即最多只能容纳 (123×123) 的方阵或相应大小的数据,而程序本身尚需占用部分内存,故内存中能容纳的数据量总是小于以上数值的。因此,在IBM-PC微机上无法按常规进行二维电法有限元正演计算,除非采用一些特别措施。

一、动态数组 (REDIM语句)

二维构造电阻率法有限元正演程序采用Quick BASIC语言编写。Quick BASIC语言是美国Microsoft公司为IBM个人计算机及兼容机提供的程序设计语言。它与BASIC语言编译完全兼容,用解释BASIC编辑的源程序只要是ASCⅡ码格式,可以不加修改或稍加修改后,就可在Quick BASIC里直接运行。采用Quick BASIC的最大优点是可以扩大微机内存。试验表明,在IBM-PC微计算机上使用Quick BASIC语言其内存的最大工作空间可达384KB,是使用BASIC语言时内存工作空间的6.4倍,而且单个数组容量可达64KB。尽管Quick BASIC提供了较大的内存空间,但对数据量庞大的二维电法有限元法计算仍显不足。为了更合理有效地利用有限的微机内存,我们在编程中使用了动态数组(REDIM语句),从而提高了微机内存的利用率。

在二维电法有限元程序设计中,一些数组是贯穿整个计算过程的,而另一些数组仅在程序中局部使用。一般来说,程序中所需要的数组无论是全局性的还是局部性的,一经定义则数组空间在编译时即得到分配,即使局部性数组在以后程序段中不再使用,但该数组在内存中仍占有编译时所分配给它的空间,这无疑是一种内存的浪费。采用动态数组则可避免这种浪费。动态数组空间是在程序运行时分配的,即动态数组使用的内存直到它需要时才被分配。对于局部性数组,在该数组使用完后,可用ERASE语句将其消除,消除后的数组在内存中不再占有空间。若该数组在以后程序段中仍需使用,只是数组的大小需要改变,则可以在用ERASE语句消除该数组后,用REDIM语句对该数组的空间进行重新定义。对于全局性数组,可采用静态数组(DIM语句)定义,静态数组和动态数组在程序中分别用元命令\$

STATIC和\$ DYNAMIC说明。现举例说明动态数组的使用方法:

```
REM $STATIC
DIM A(50, 50)    定义静态数组(全局数组)A
:
REM $DYNAMIC
REDIM B(40, 40), C(60, 60)  定义动态数组B和C
:
ERASE B, C      消除动态数组B和C
REDIM B(120, 120) 重建定义动态数组B
:
```

显然, 动态数组的使用是提高微机内存利用率的一种有效手段。

二、刚度矩阵的一维变带宽压缩存储

应用有限单元法求解双点电源二维构造电场的变分问题

$$\left\{ \begin{array}{l} J(\varphi) = \frac{1}{2} \int \int \left\{ \sigma(x, z) [(\nabla \varphi \cdot \nabla \varphi) + \lambda^2 \varphi^2] - \right. \\ \left. 2 \sum_{i=1}^n I_i \delta(x-x_i) \delta(z-z_i) \varphi \right\} dx dz = \text{极值} \dots\dots\dots (1) \\ \varphi|_{\Gamma_2} = \frac{1}{\pi} [I_1 K_0(\lambda r_1) + I_2 K_0(\lambda r_2)] \cdot \sum_{i=1}^n W_i \rho_i \end{array} \right.$$

可建立如下形式的有限元方程

$$[K][\varphi][G] \dots\dots\dots (2)$$

在方程(1)和(2)中

- σ ——岩石导电率;
- λ ——离散波数;
- δ ——狄拉克函数;
- I ——供电电流强度;
- Γ_2 ——求解域D除地表以外的边界;
- r_1, r_2 ——点电源 I_1 和 I_2 到边界 Γ_2 的距离;
- K_0 ——零阶变型贝塞尔函数;
- n ——D域中不同电阻率介质的总数;
- W_i ——与D域中各种电性介质有关的系数;
- $[K]$ ——刚度矩阵;
- $[\varphi]$ ——由有限元网格节点转换电位 φ 组成的列矩阵;
- $[G]$ ——与点电流源有关的列矩阵。

在有限元方程(2)中, 刚度矩阵 $[K]$ 的阶数通常是很高的, 它对微机内存的需求量非常大, 也是整个二维电法正演计算中微机内存占有量最大的一部分。因此, 如何存储刚度矩阵 $[K]$ 便成为节省微机内存的关键所在。

根据刚度矩阵对称正定、带状、稀疏的特点, 我们采用一维变带宽压缩存储方式存矩阵 $[K]$, 从而使刚度矩阵对微机内存的需求量大大降低。下面举例说明其基本原理。

我们采用三节点三角形网格(图1)对求解域D进行离散, 其刚度矩阵中的元素分布情况如图2所示。由图1看出, 节点4包括本身在内共有5个相关节点。因此, 在图2所示的

刚度矩阵中，第四行仅有 5 个非零元素。一般地说，在二维电法问题中，一个节点的相关节点不会超过 7 个。若网格中有 700 个节点，则刚度矩阵中任意一行中的非零元素个数与该行元素总数之比值不大于 1/100，即稀疏度在 1% 以下。网格节点数愈多，刚度矩阵愈稀疏。

一维变带宽压缩存储的基本思想是：只存刚度矩阵中各行（或各列）的第一个非零元素到主对角线元素之间的元素。根据这一原则，可将图 2 转换为图 3，并将图 3 中带宽变化线以内（即两条粗黑折线以内）的元素压缩成一维数组的形式存入微机，其存储顺序如图 4 所示。为便于解方程时寻找出相应的矩阵元素，程序中须有反映各元素在一维数组中地址码的信息。实际上，根据刚度矩阵的特点以及节点的相关性与不相关性，只要记录主对角线元素在一维数组中的存储序号，其余元素便可根据相应的关系式寻找其地址。主对角线元素的存储序号如图 5 所示。

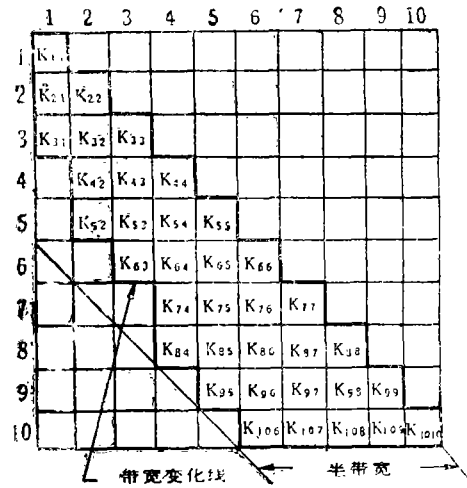


图 3 变带宽刚度矩阵示意图

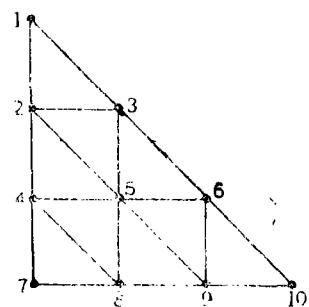


图 1 有限元网格示意图

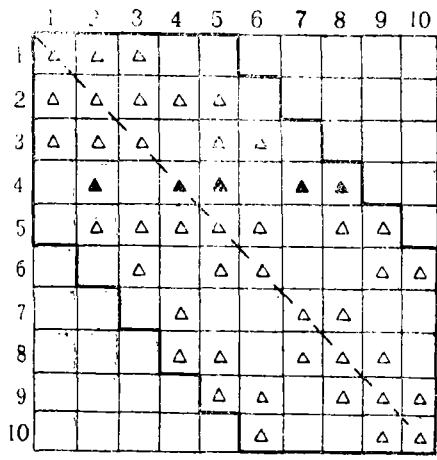


图 2 刚度矩阵元素分布示意图



图 4 矩阵元素在一维数组中的存放顺序

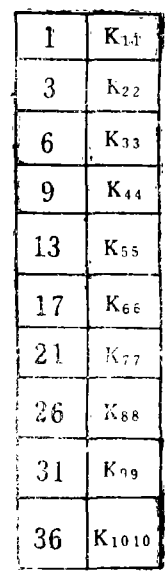


图 5 对角线元素在一维数组中的序号

表 1 给出了几种不同存储方法对微机内存需求量的对比结果，可以看出，对于同一刚度矩阵一维变带宽压缩存储所需内存空间仅占原矩阵的百分之几。实践表明，网格节点数愈多，效果愈明显。

几种不同存储刚度矩阵方法对比结果 表 1

四格节点	存储方法	占内存单元数	与全矩阵的百分比
13 × 7	全矩阵存储	3281	100%
	半矩阵存储	4186	50.5%
	一维变带存储	721	8.7%
	全矩阵存储	408321	100%
71 × 9	半矩阵存储	204430	50.1%
	一维变带存储	6597	1.6%

三、强加边界条件的处理方法

在解有限元方程(2)之前,必须对强加边界条件进行处理,即将边界值综合到有限元方程中去。

将式(1)中的边界条件简记为

$$\varphi|_{\Gamma_2} = f(x, z) \dots \dots \dots (3)$$

并设节点 m 位于边界 Γ_2 上,则节点 m 上的 φ 值为

$$\varphi_m = f(x_m, z_m) = f_m$$

将方程(1)展开为

$$\begin{cases} K_{11}\varphi_1 + K_{12}\varphi_2 + \dots + K_{1m}\varphi_m + \dots + K_{1n}\varphi_n = G_1 \\ K_{21}\varphi_1 + K_{22}\varphi_2 + \dots + K_{2m}\varphi_m + \dots + K_{2n}\varphi_n = G_2 \\ \vdots \\ K_{m1}\varphi_1 + K_{m2}\varphi_2 + \dots + K_{mm}\varphi_m + \dots + K_{mn}\varphi_n = G_m \\ \vdots \\ K_{n1}\varphi_1 + K_{n2}\varphi_2 + \dots + K_{nm}\varphi_m + \dots + K_{nn}\varphi_n = G_n \end{cases} \dots \dots \dots (4)$$

式中: n 为节点总数。

通常对强加边界条件的处理方法是将(4)式中各方程等号左边的第 m 列元素 $K_{1m}\varphi_m$ 、 $K_{2m}\varphi_m$ 、 \dots 、 $K_{nm}\varphi_m$ 移到方程组该行的等号右端,并将 $\varphi_m = f_m$ 代入方程组,然后将方程组中的第 m 个方程式删去,使(4)式变为

$$\begin{cases} K_{11}\varphi_1 + K_{12}\varphi_2 + \dots + K_{1n}\varphi_n = G_1 - K_{1m}f_m \\ K_{21}\varphi_1 + K_{22}\varphi_2 + \dots + K_{2n}\varphi_n = G_2 - K_{2m}f_m \\ \vdots \\ K_{n1}\varphi_1 + K_{n2}\varphi_2 + \dots + K_{nn}\varphi_n = G_n - K_{nm}f_m \end{cases} \dots \dots \dots (5)$$

经过上述处理后,刚度矩阵的阶数虽然有所降低,但是由于这种处理方法改变了原刚度矩阵元素的排列次序,因而必须重新计算新刚度矩阵在一维变带宽压缩存储中所需的单元数。实践证明,这样处理将耗费大量机时。例如,对于700个节点的网格,在IBM-PC微型机上进行上述处理需15分钟。为了提高计算效率,我们对强加边界条件进行了特殊处理,原理如下:

将(4)式变换成下列形式

$$\begin{cases} K_{11}\varphi_1 + K_{12}\varphi_2 + \dots + 0 + \dots + K_{1n}\varphi_n = G_1 - K_{1m}f_m \\ K_{21}\varphi_1 + K_{22}\varphi_2 + \dots + 0 + \dots + K_{2n}\varphi_n = G_2 - K_{2m}f_m \\ \vdots \\ 0 + 0 + \dots + \varphi_m + \dots + 0 = f_m \\ \vdots \\ K_{n1}\varphi_1 + K_{n2}\varphi_2 + \dots + 0 + \dots + K_{nn}\varphi_n = G_n - K_{nm}f_m \end{cases} \dots \dots \dots (6)$$

其矩阵形式为

$$\begin{bmatrix} K_{11} & K_{12} & \cdots & 0 & \cdots & K_{1n} \\ K_{21} & K_{22} & \cdots & 0 & \cdots & K_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ K_{n1} & K_{n2} & \cdots & 0 & \cdots & K_{nn} \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_m \\ \vdots \\ \varphi_n \end{bmatrix} = \begin{bmatrix} G_1 - K_{1m}f_m \\ G_2 - K_{2m}f_m \\ \vdots \\ f_m \\ \vdots \\ G_n - K_{nm}f_m \end{bmatrix} \cdots \cdots (7)$$

由(7)式可看出,当处理节点 m 上的强加边界条件时,只需将刚度矩阵中的对角线元素 K_{mm} 置1,将第 m 行和第 m 列的其它元素全部置零,并将第 m 个方程式的右端项改为强加边界值 f_m ,同时在其余各方程式的右端分别减去该节点的强加边界值与未处理前对应第 m 列元素的乘积即可。若强加边界面上有 l 个节点,则如此循环处理 l 次。由于这种处理方法没有改变原刚度矩阵元素的排列次序,仅改变了部分元素值,因而对强加边界条件处理后不需要重新计算新刚度矩阵在一维变带宽压缩存储中所需的单元数,只需将原一维数组中的部分元素值置换即可,从而减少了计算量。例如,在IBM-PC微型机上同样处理700个网格节点,前一种需15分钟,而后一种方法所需时间不足1分钟,其效率是明显的。

四、结 束 语

经在IBM-PC微计算机上试算表明,在二维电法有限元正演计算中,采用本文介绍的三种措施提高了微机内存的利用率,并减少了运算量,从而使二维电法有限元正演计算的微机化成为可能。文中介绍的方法虽然是以二维电法为例讨论的,但同时也适合其它工程问题有限元算法,因而具有较普遍的意义。

参 考 文 献

- (1)J.E.AKIN: *Application and Implementation of Finite Element Method*, Academic Press INC, New York, 1982.
- (2)E.歇顿, D.R.J.欧文:《有限元程序设计》,新时代出版社,1982.
- (3)张伟均、韩瑞雪译:《Microsoft Quick-BASIC 编译程序实用指南》,轻工业出版社,1988.

推覆和滑脱构造研究硕果累累

由第一煤田地质勘探公司、广东煤田地质公司、湖南煤田地质公司和安徽煤田地质公司分别承担,中国矿业大学协作的《安徽省淮南煤田颍凤推覆构造与找煤研究》、《广东省兴梅煤田滑脱构造与找煤研究》、《湖南省涟邵煤田北段测水煤系滑脱构造的研究与找煤预测》以及《安徽两淮煤田推覆构造与找煤研究》四个课题,于1989年10月在湖南株洲市同时通过评审。

这四个课题应用现代地质科学理论和多

种先进手段,对本区广泛存在的断层构造的形态特征、分布范围、形成机制以及对煤系、煤层赋存的影响,进行了比较系统的研究,并取得显著的社会效益和经济效益。安徽省淮南煤田颍凤区在推覆体下发现新的煤炭资源,总储量达 $21.49 \times 10^8 t$;广东兴梅煤田获预测储量 $1.16 \times 10^8 t$;湖南涟邵煤田北段五个测区预测储量 $2.8 \times 10^8 t$;安徽两淮煤田七个找煤预测区预测煤储量达 $24.48 \times 10^8 t$ 。

(蒋士钧)