

平面任意区域自适应网格逼近^{*1)}

王建华

(上海交通大学)

王卫中

(西南电力设计院)

ADAPTIVE MESH APPROXIMATION FOR ARBITRARY PLANAR REGIONS

Wang Jianhua

(Shanghai Jiao Tong University)

Wang Weizhong

(South West Electronic Power Design Institute)

Abstract

A Quadrilateral / Triangular mesh generator suitable for planar regions is implemented in this paper, the mesh generator can be used to refine arbitrary planar regions adaptively.

1. 前言

实现自适应有限元的关键技术之一是研制出网格生成器。一般而言, 网格生成器必须具有对任意曲线构成的封闭区域进行加密直到满足预先给定的精度要求。同样对具有孔洞及复杂形状的边界, 网格生成器亦应达到预先要求加密的精度。具备上述功能后, 方能在自适应过程中依据误差大小, 达到满足指定误差要求的目标。

自适应网格生成器研究较早且相对成熟的是三角形单元。三角形单元的优势在于对复杂边界的适应能力强, 实现局部加密功能较四边形单元容易。四边形单元处理局部加密问题时不可避免地要遇到不协调点(即四边形某一边上出现三个结点的现象)问题, 从已发表的文献看, 对不协调点的处理均是探讨性的。

* 1994年7月25日收到。

1) 国家自然科学基金、国家博士后基金资助项目。

对自适应加密过程中所形成的一系列网格,用什么方法进行求解也是人们非常关心的问题.从文献中看主要有两类方法:其一是直接法,对系列网格的结点进行重新编号,得到优化的带宽便于直接法求解;其二是迭代法,使用迭代法无须进行带宽优化,且非零元素可以不存,故可节省内存.由于常规迭代法(如GS, SOR等)求解速度较慢,因此,采用较多的是各类加速迭代法(如共轭斜量加速Jacobi迭代法等).近年来发展起来的多重网格迭代法引起了人们的极大关注.由于多重网格法的实现要求有相互嵌套的网格,而自适应过程形成的正好是一系列相互嵌套的网格,人们很自然想到将两种技术结合起来.文献[1]的研究结果表明:多重网格法与共轭斜量加速Jacobi迭代法求解同一问题时,多重网格法比共轭斜量加速Jacobi法快十多倍.此外,用多重网格法求解方程还有其它一些优点,如文献[2]中用多重网格迭代法求解严重病态方程获得令人满意的结果.自适应网格加密技术与多重网格迭代法相结合,对数据结构的设计有较高的要求,程序的实现也有一定的难度.有关这方面的详细内容可参考文献[1, 3].

本文基于文献[4]和文献[5]提出的数据结构并作了一定的修改后,以直线逼近曲线的基本方法进行三角形与四边形自适应网格生成器的程序实现.三角形单元同时完成了多重网格法求解技术,四边形单元与多重网格相结合的成果将另文发表.

2. 数据结构

2.1 分子表结构(Molecular List Structure)^[4]

设 τ 为任意的协调三角形剖分(即 τ 中相交的和非同等的三角形之间只能有公共边或公共顶点),其顶点的集合为 $N(\tau)$,对于每个顶点 $J \in N(\tau)$,定义整形函数 $n(j)$,表示与结点 J 直接相连的顶点个数.如图1中的顶点7, $n(7)=5$.其次定义一个双值函数 $b(j)$

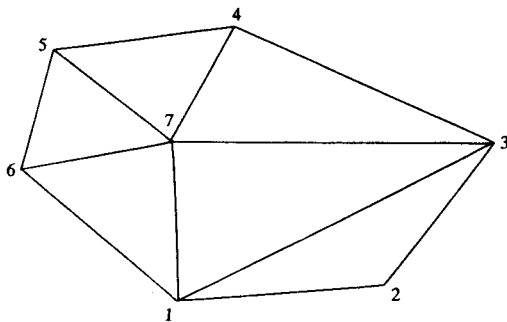


图1 协调三角形剖分

$$b(j) = \begin{cases} 0, & \text{若 } j \text{ 是内点,} \\ 1, & \text{若 } j \text{ 是边界点.} \end{cases}$$

如图1中, $b(1)=1$, $b(7)=0$.

分子表中的每个分子 J 定义为一个多边形,多边形区域内部仅含结点 J .与 J 相关的分子表为

$$M(j) = (j_1, j_2, \dots, j_{n(j)}),$$

式中 $j_1, j_2, \dots, j_{n(j)}$ 为分子 J 的顶点.若 $b(j)=1$,则 $b(j_1)=1$ (若 j 为边界点,则 j_1 必为边界点).如图1所示. $M(7)=(5,6,1,3,4)$, $M(4)=(5,7,3)$.

一个协调的三角形剖分 τ 可表示为

$$L(\tau) = (M(1), M(2), M(3), \dots, M(N)),$$

其中 N 为 $L(\tau)$ 中元素个数,三角形集合可按下面顺序生成:

$$\tau = \bigcup_{j=1}^N \{T(j, j_k, j_{k+1}), k=1, 2, \dots, n(j)-1, \\ j_k, j_{k+1} \in M(j), j_k > j, j_{k+1} > j\} \cup V_j, \\ V_j = \begin{cases} T(j, j_{n(j)}, j_1), & \text{若 } b(j)=0 \text{ 及 } j_{n(j)} > j, j_1 > j, \\ 0 & \text{, 其它,} \end{cases}$$

其中 $T(i, j, k)$ 为顶点是 i, j, k 的三角形.

程序实现中涉及的主要变量如下:

LEVEL (k) —— 指示在第 k 层网格中, 新增或修改过的分子存贮(按增序方式)在数组 NMODIF 中的位置.

NMODIF (LEVEL(k) + j) —— 在第 k 层网格中新增或修改过的第 $j+1$ 个分子的序号.

NCONE(i) —— 分子 NMODIF(i) 的顶点个数.

IPTER (i) —— 指示分子 NMODIF(i) 中的顶点存贮在数组 NEINOD 中的位置.

NEINOD (IPTER (i) + l) —— 分子 NMODIF (i) 的第 $l+1$ 个顶点.

COORX, COORY —— 结点的 X, Y 坐标数组.

2.2 四边形单元数据结构^[5]

本文所采用的数据结构如下:

NODES(j, NEL) —— 单元 NEL 结点 j 全局号.

XCO (JCO, NODE) —— 结点 NODE 的全局坐标 JCO.

NELCON (NC, NEL) —— 单元 NEL 的第 NC 个邻接单元, NC=1, ..., 8.

LEVEL (NEL) —— 单元 NEL 加密的层数.

NNODE —— 总的结点数.

NELEM —— 网格中的总的单元数.

NELGRP (i, IGR) —— IGR 单元组中第 i 个单元. 若大于零, 表示一个单元号; 若小于零, 表示一个单元组.

3. 网格加密算法与误差估计方法

3.1 三角形局部加密算法

设 T_0 是任意一个初始协调三角形剖分, 并设 $\bar{T}_0 \subset T_0$ 是要求被细化的三角形集, 则一个新的、局部细化的三角形剖分可按下述步骤获得:

a. 对每个三角形 $T \in \bar{T}_0$, 用最长边中点进行剖分. 令 T'_0 是这方式下产生的三角形剖分, 若 T'_0 是一个协调三角形剖分, 则已完成, 否则作步骤 b.

b. 找出步骤 a 中所有非协调的三角形(三角形某一边出现三个结点)形成的集合 $R_1 \subset T'_0$.

c. 置 $k=1$.

d. 对每个三角形 $T \in R_k$, 设其非协调点(三角形单元某一边出现三个结点的中间结

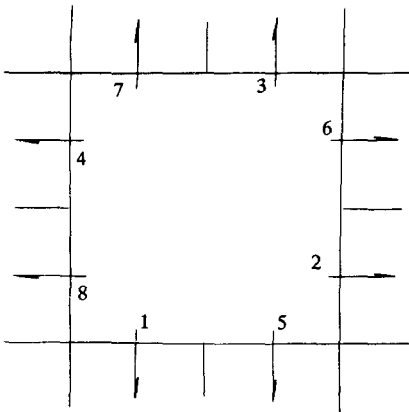


图2 单元A的连接

点)是 P , 用 T 的最长边的中点剖分 T ; 若 P 并非 T 的最长边的中点, 再用最长边中点联结 P .

e. 令 T_0^k 是经步骤 d 后所产生的三角形剖分, 找出其中的所有非协调的三角形集合 $R_{k+1} \subset T_0^k$.

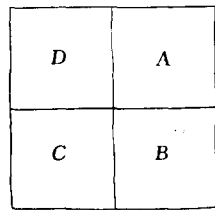
f. R_{k+1} 是否空集? 是, 停止, 获得协调三角形剖分 $T_1 = T_0^k$; 否, 置 $k = k + 1$, 转向 d .

显然, 三角形局部加密算法是有限的.

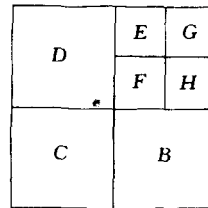
3.2 四边形局部加密算法

图2所示为一个四边形单元与周围邻接单元的编号规则, 且规定四边形单元的每条边不能多于二个单元与之相连.

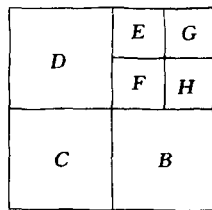
设我们从一个均匀的初始网格开始细化(图3(a)), 单元 A 细化的结果见图3(b). 若需细化单元 H (图3(c)), 由于已规定每条边至多只能与两个单元相连, 因此 B 单元必须先细化, 细化结果见图3(d).



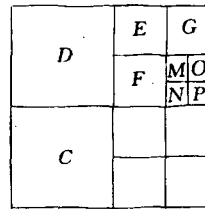
(a)



(b)



(c)



(d)

图3 四边形单元细化

由于我们不能知道为细化某个特定单元而须先细化的单元号, 故引入下面的递归算法:

SUBROUTINE DIVIDE (NEL1, NEL2)

NEL1: (输入) 须细化的单元号.

NEL2: (输出) = NELO 若 NELO 在细化 NEL1 前须先细化.

有了子程序 DIVIDE 后, 算法如下:

```

SUBROUTINE REFINE (NELR)
REPEAT
NEL1 = NELR
CALL DIVIDE (NEL1, NEL2)
WHILE (NEL1. NE. NEL2)
NEL1 = NEL2
CALL DIVIDE (NEL1, NEL2)
END WHILE
UNTIL (NEL2. EQ. NELR)

```

当一个单元确定要细化时, 首先产生新的单元号、结点号、结点坐标、结点总数, 然后形成新单元的 NELCON 数组, 再修改邻接单元的 NELCON 数组. 方法如下:

新单元的 NELCON 数组可通过老单元的 NELCON 数组形成. 通过将老单元改成由老单元新生成的两个单元以实现邻接单元的修改. 由下述条件可判断不协调点. 若 $NELCON(i) > 0$ 且 $NELCON(i+4) > 0$, 则 i 单元必有不协调点.

退化一个特定单元 NEL, 可按以下步骤进行:

1. 找到该单元在 NELGRP 数组中的位置, 假设在 $NELGRP(i)$ 中.
2. 判断该 NELGRP 是否可退化(指回退到加密前的状态), 其标准为:
 - ① $NELGRP(i, 0)$ 不能为零, 为 0 表示是初始层网格, 不可退化;
 - ② $NELGRP(i)$ 中含有单元组元素(即有值为负的元素), 应先退化该单元组;
 - ③ 邻接单元中有网格层次比该单元高的应先退化层次高的单元.

重复上述过程, 直到 $NELGRP(i)$ 可退化为止. 退化一个单元所用的变量有:

CLEV: 当前层数,

NNELGRP + 1: 指向下一个可用的 NELGRP 数组.

NNELGRP: 指向当前最后一个 NELGRP 数组.

NNELGRP($i, 0$): 指向父结点(上一层网格单元结点), 等于 0 表示无父结点(初始网格结点).

$$\left\{ \begin{array}{l} NNELGRP(i, 1) \\ NNELGRP(i, 2) \text{ 同时指向父结点的单元号, 子结点.} \\ NNELGRP(i, 3) \\ NNELGRP(i, 4) \end{array} \right.$$

子程序如下:

```

SUBROUTINE UNREFINE (Group, ULEV)
IF(ULEV. EQ. 0)
按组退化
IF(ULEV. NE. 0)
按层次退化.

```

3.3 不协调点的处理

本文采用在单元形心处增加结点的方法进行处理, 与其它近似处理方法相比, 本文

方法的不足之处是增加了自由度, 但该方法实现简单且计算精度有所提高。

四边形单元不协调点的出现只有三种情况, 其一是四边形只有一条边上出现不协调点; 其二是二条边有不协调点; 其三是有三条边上出现不协调点。图4所示为三种不同情况。图5是本文采用的处理方法, 每次网格加密后, 进行有限元计算之前对不协调点进行处理, 四边形单元与三角形单元分别形成单刚。有限元求解结束后将网格恢复到未进行处理的原状并继续进行四边形网格加密, 此过程不断重复直至计算结束。

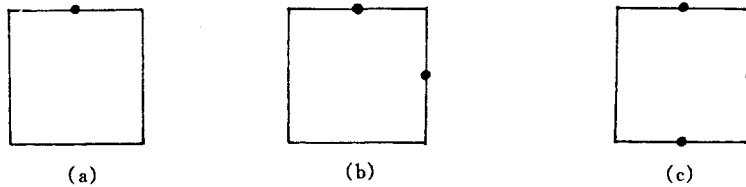


图4 三种不协调点

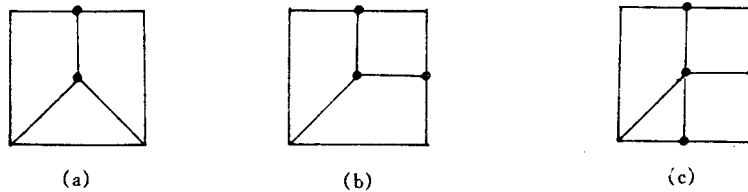


图5 不协调点处理方法

3.4 加密指定区域方法

对由任意曲线构成的封闭区域加密或对具有孔洞及复杂形状的边界进行加密, 均要求网格生成器能对指定区域进行加密直至满足预先给定的要求。本文采用下述步骤进行指定区域的加密:

1. 在输入初始网格数据同时输入控制参数, 对规则曲线(如圆、椭圆等)输入其几何方程; 对任意曲线以折线进行逼近并输入相应的直线方程; 对参数不同的两条曲线交点须输入特殊标志。

2. 加密网格时, 若遇规则曲线边界, 将新增结点的坐标按其曲线方程取值。若遇任意边界, 新增结点的坐标按相应的直线方程取值。

3. 加密指定区域(如以某结点为圆心加密一个圆域或加密某一封闭曲线围成的区域)时, 网格加密一次后, 对预先给定的要求(如规定最大加密层数或单元的最小几何尺寸)进行判断, 若满足要求, 停止加密, 否则, 继续在区域内加密(区域外按算法规定进行协调), 直至满足要求为止。

3.5 误差估计方法^[6]

由有限元解出的近似位移 u_h 可求出近似应力 $\sigma_h = DSu_h$, 此时应力的局部误差

$$e_\sigma = \sigma - \sigma_h,$$

其中 D 为弹性矩阵, S 为应变矩阵, σ 为应力的精确解。用光滑应力 $\bar{\sigma}$ 替换 σ 可得下式

$$e_\sigma = \bar{\sigma} - \sigma_h = \hat{e}_\sigma,$$

式中 $\bar{\sigma} = N\hat{\sigma}$, N 为形函数, $\hat{\sigma}$ 为绕结点平均应力. 求 $\hat{\sigma}$ 的方法主要有投影法、绕结点平均法. 投影法由于要求一个与原问题同规模的方程, 计算量增大较多, 而绕结点平均法实现简单且节约时间, 故本文采用绕结点平均法.

单元的误差定义为

$$\|\hat{e}_\sigma\|_i^2 = \int_{A_i} [(\sigma_x - N\hat{\sigma}_x)^2 + (\sigma_y - N\hat{\sigma}_y)^2] dA,$$

式中 A_i 为第 i 个单元的面积. 全局(整个计算区域)误差的定义为

$$\|e_\sigma\|^2 = \sum_{i=1}^{NE} \|\hat{e}_\sigma\|_i^2,$$

式中 NE 为单元总数. 应力计算相对误差的定义

$$\eta' = [\|e_\sigma\|^2 / \|\sigma_h\|^2]^{1/2},$$

式中 $\|\sigma_h\|^2 = \int_{\Omega} [(N\sigma_x)^2 + (N\sigma_y)^2] d\Omega.$

若计算误差以 η 表示, 给定的误差用 η_{\max} 表示, 则计算退出的条件为

$$\eta \leq \eta_{\max}.$$

假设误差在各个单元之间平均分布, 则上式为

$$\|\hat{e}_\sigma\|_i \leq \eta_{\max} [\|\sigma_h\|^2 / NE]^{1/2} = e_{\max},$$

式中 e_{\max} 为单元误差最大容许值, 定义

$$\xi_i = \|\hat{e}_\sigma\|_i / e_{\max}$$

为加密准则, 若 $\xi_i > 1$, 该单元应加密.

有关三角形单元多重网格算法描述及程序实现可参考文献[1,3].

4. 网格算例

4.1 三角形单元

本文研制的三角形网格生成器具有下述功能: 可按指定的最小单元尺寸(单元的最短边)加密角域、圆域、带状域及由圆和直线组成的任意区域. 在记录网格加密“历史”信息的基础上, 可按层退化网格. 有关三角形单元退化算法及算例可参见文献[7]. 稍后将要提到的三角形单元与多重网格法结合的算例, 是其另一重要功能.

图 6 所示为同济大学英文缩写 T. J. U 的初始网格图(人工输入). 该网格有 139 个结点, 指定的加密区域有: 8 个角域, 2 个圆域, 3 个带形区域. 指定的最小网格尺寸为 0.01 米. 对所指定的加密区域, 为满足区域内单元最小尺寸小于 0.01 米这一条件, 我们采用了强迫加密措施, 即在进一步加密前, 先判断区域内单元最小尺寸是否满足要求? 是, 退出. 否, 继续在区域内加密, 直到满足要求为止.

图 7 为网格加密的最后结果. 网格内的总结点数为 6561 个, 图 7 表明, 无论是角域、圆域还是带状区域加密效果令人满意.

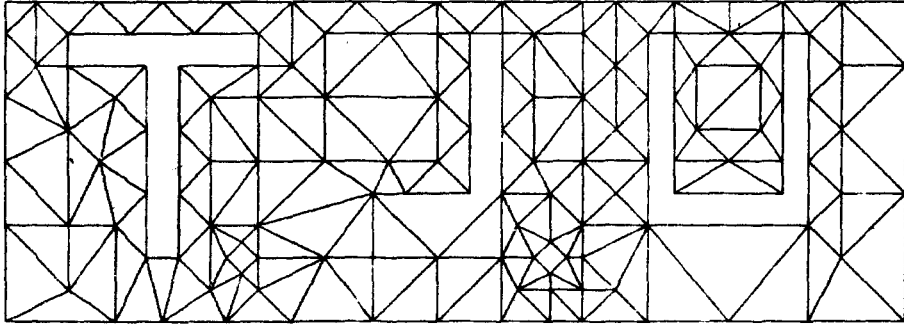


图6 T. J. U 初始网格

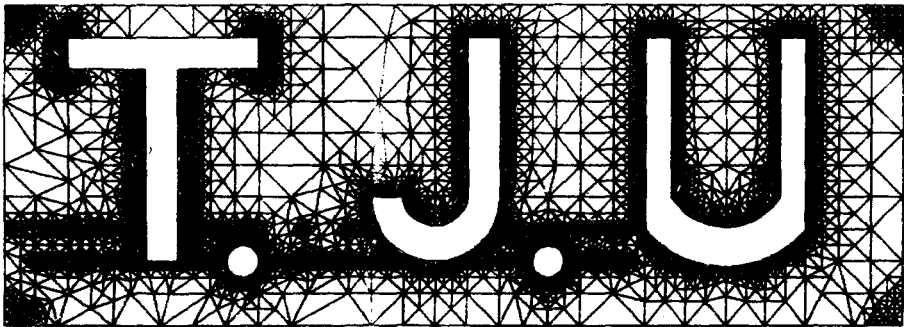


图7 T. J. U 最后一层网格

4.2 四边形单元

图8所示为一圆环内连接一小圆的初始网格图. 小圆与圆环的交点属于二个参数不

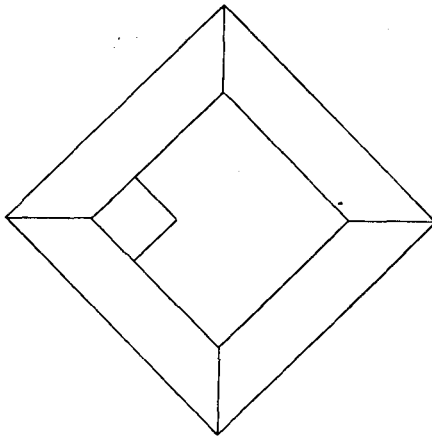


图8 圆环初始网格

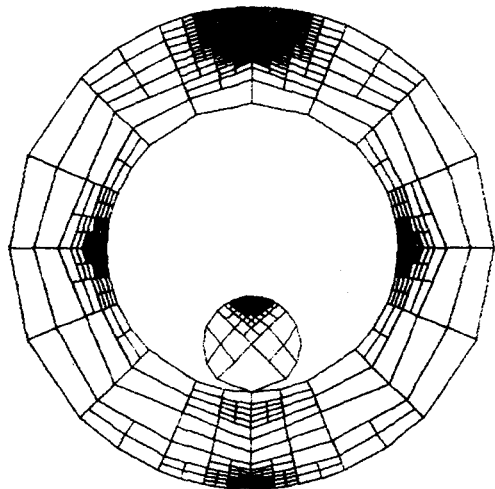


图9 圆环计算结果

同的圆，对这类点的处理，我们分别给出指示两个圆的标识，以便网格加密时作出正确判断。初始网格 11 个结点，指定五个区域加密，计算结果见图 9，结点数为 4473 个。

类似的情况在工程计算中常常遇到，为说明网格生成器处理这类问题的有效性，再举吊钩的网格加密算例。初始网格见图 10，结点数为 24 个，在每组曲线的交点处设置控制点，并指定三个局部加密区域，经过五次连续加密达到精度要求，第五层网格的计算结果见图 11，该图中共有节点 2347 个。

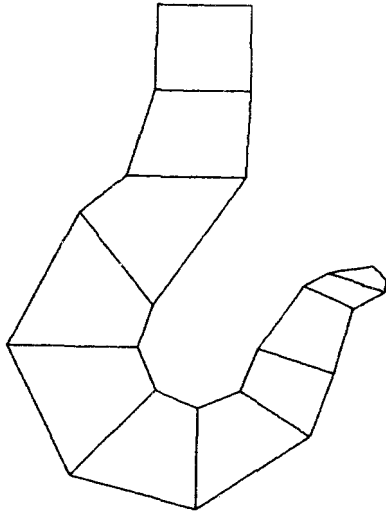


图 10 吊钩初始网格

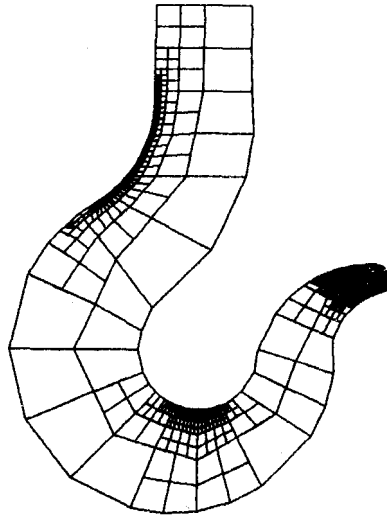


图 11 吊钩计算结果

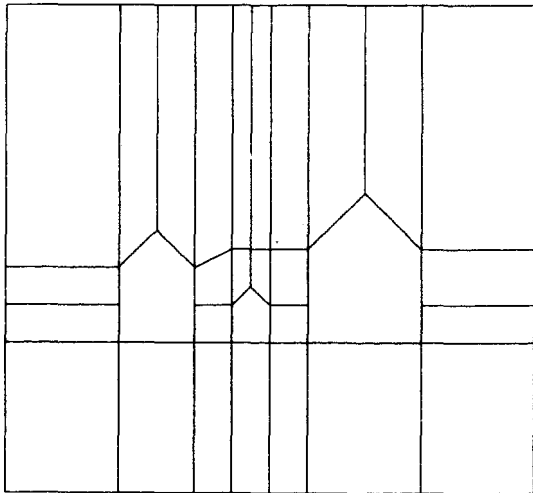


图 12 洞室初始网格

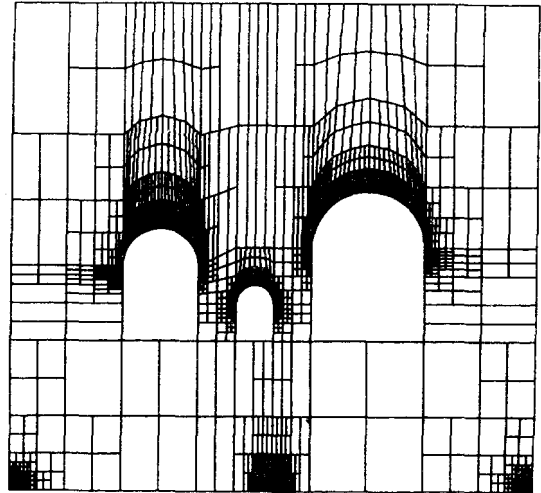


图 13 洞室计算结果

四边形单元的第三个算例为一个工程实例，三个洞室同时开挖。在洞室周围是应力集中区，指定为加密区域。初始网格见图 12，除了三个洞室外，还指定了三个局部加密区域。图 12 中的结点数为 47 个。经过八次加密达到精度要求，计算结果见图 13，在

该图中单元数为 15915 个, 结点数为 17500 个.

4.3 自适应多重网格算例

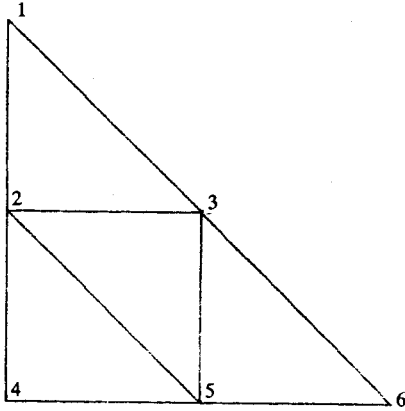


图 14 第一层 初始网格

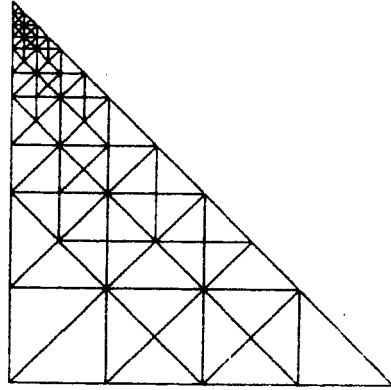


图 15 第十层 结点数 61 个

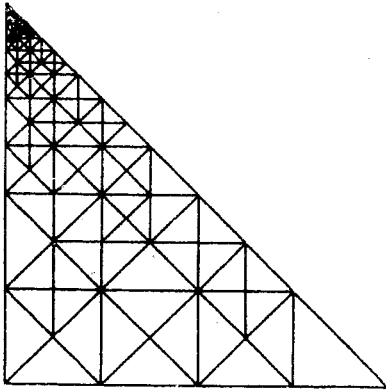


图 16 第二十三层 结点数 194 个

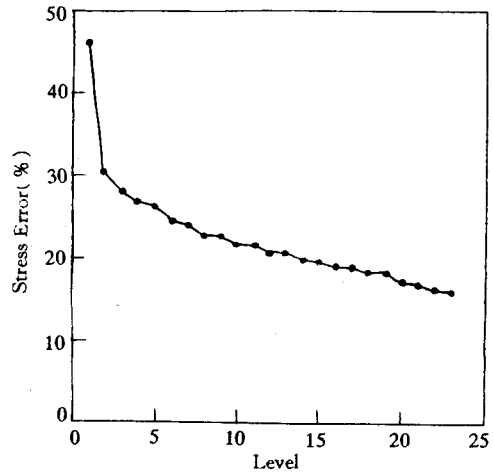


图 17 误差收敛图

为说明自适应网格加密技术与多重网格迭代法相结合的完整结果, 现举一个简单算例, 该算例的基本数据取之文献[8]. 从对角受压的正方形薄板中取出四分之一部分(见图 14), 荷载作用在 1 号结点. 取应力计算控制精度 $\eta_{\max}=16\%$, 经 23 层加密后, 应力计算相对误差 $\eta=15.8\%$, 由于 $\eta < \eta_{\max}$, 计算结束. 图 15 和图 16 分别为第十层和第二十三层的网格图. 图 17 为应力计算误差收敛图.

5. 结束语

实际工程的边界形状往往十分复杂, 对复杂边界组成的计算区域进行自适应分析, 对网格生成器有较高的要求. 计算结果表明, 本文研制的网格生成器可对任意曲线组成

的计算区域进行自适应加密. 网格生成器与多重网格法的有机结合, 为大规模高精度的工程计算提供了一个有力的工具.

参 考 文 献

- [1] 王建华, 线弹性有限元的自适应加密与多重网格求解, 河海大学学报, 3(1994) 16— 22.
- [2] 王建华, MG法求解严重病态方程, 力学与实践, 1(1995) 59 — 60.
- [3] 王建华, 同济大学博士论文, (1992).
- [4] M. C. Rivara, Algorithms for refining triangular grids suitable for adaptive and multigrid techniques, Int. J. Numer. methods Eng., Vol.20 (1984) 745— 756.
- [5] O. C. Zienkiewicz, J. Z. Zhu, A simple error estimator and adaptive procedure for practical engineering analysis, Int. J. Numer. Methods Eng., Vol. 24 (1989) 337 — 357.
- [6] P. Devloo, J. T. Oden, T. Strouboulis, Implementation of an adaptive refinement technique for the SUPG algorithms, Computer Methods in Applied Mechanics and Engineering, Vol.61 (1987), 339— 358.
- [7] 王建华, 平面任意区域三角形网格自动生成, 计算机应用与软件, 1(1995), 26 — 32.
- [8] 华东水利学院, 弹性力学问题的有限单元法, 修订版, 水利出版社, 1978.