

文章编号:1000-582x(2000)01-0028-04

数值流形方法物理覆盖系统的自动剖分

⑧
28-31, 44

张湘伟¹, 蔡永昌², 廖林灿²

TU452

(1. 汕头大学 科学院力学研究所, 汕头 515063; 2. 重庆大学 汽车摩托车工程中心, 重庆 400044)

TU43

摘要: 利用面向对象设计思想将岩土力学中数值流形方法的物理覆盖系统抽象为独立的数据类, 在此基础上完成了物理覆盖系统的自动剖分算法。对裂缝切割的连续材料, 在裂缝的扩展过程中能自动生成正确的流形单元。算例表明, 本算法是可靠和有效的。

关键词: 流形方法; 连续材料; 覆盖; 裂纹

中图分类号: TU 452; TB 115

文献标识码: A

岩土力学 数值流形法

物理覆盖系统, 自动剖分

在有限覆盖基础上新发展的数值流形方法是一种通用数值方法, 它统一解决有限元、DDA 和解析法的计算问题, 在岩土工程中已获得了较为广泛的应用。数值流形方法能够计算块体和裂缝大变形的位移, 也能进行连续材料的有限元法计算。

将数值流形方法用于连续材料的裂缝计算时, 可以将有限元网格转化为流形方法的有限覆盖(数学网格), 裂缝则作为新的物理网格输入。作为数学覆盖的有限元自动剖分理论研究很多, 已形成比较成熟的理论。但是对于连续材料的物理覆盖的划分, 由于裂缝切割的任一流形单元都有完全不同的节点或物理覆盖码, 文[1]用于块体的节理网格拓扑生成技术不再适用。目前大都采用在裂纹的扩展过程中人工干预输入新的物理网格的办法^[2], 这种办法不仅费时繁琐, 而且容易造成错误和影响运算速度。作者采用面向对象的设计方法^[3-6], 把数值流形方法的物理覆盖系统划分成一些相互独立的数据类, 在此基础上完成了物理覆盖系统的自动剖分算法。算例表明, 本算法是正确和可靠的, 对于推广流形方法的研究具有积极作用。

在图1中, 被裂缝切割的流形单元的物理覆盖列于表1。

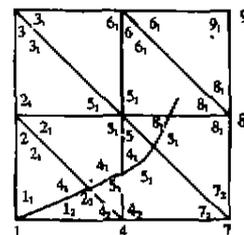


图1 有一条裂缝的物理覆盖系统

表1 流形单元的物理覆盖

原始单元	流形单元	流形单元
1, 4, 2	1 ₁ , 4 ₁ , 2 ₁	1 ₂ , 4 ₂ , 2 ₂
2, 4, 5	2 ₁ , 4 ₁ , 5 ₁	2 ₂ , 4 ₂ , 5 ₁
4, 7, 5	4 ₁ , 7 ₁ , 5 ₁	4 ₂ , 7 ₂ , 5 ₁
5, 7, 8	5 ₁ , 7 ₁ , 8 ₁	5 ₁ , 7 ₂ , 8 ₁

因此, 可以将数值流形方法的物理覆盖系统的数据划分为数学覆盖节点类 NodeM, 物理覆盖节点类 NodeU, 流形单元类 Element 和扩展中的裂缝节点类 Crack。

1.1 数学覆盖节点类

数值流形方法中的数学覆盖由用户选择, 可以将有限元的网格或级数收敛域等转换成数学覆盖网格。

1 数值流形方法的数据类

在流形方法中有两套相互独立的覆盖系统, 它们是数学覆盖和物理覆盖。物理覆盖的交集形成流形单元, 在三角形有限元情况下流形单元必是3个物理覆盖的公共区。

收稿日期: 1999-07-16

基金项目: 国家自然科学基金资助项目(9675039)

作者简介: 张湘伟(1950-), 男, 山西闻喜人, 教授, 博士, 现为汕头大学校长, 主要从事计算力学及现代设计方法研究。

数学覆盖可以移动,可以开裂并且可以随意被移开或加上,它定义了数值解的精度。

图 1 示的数学覆盖网格的任一节点,含有这一节点的所有单元形成一个数学覆盖(在几何拓扑里称为“星”),如节点 3 的数学覆盖 V3 有两个单元 3,5,6、2,5,3。

定义类 NodeM 用于记录独立的数学覆盖节点的信息,它的数据成员有节点的位移分量数 *nw*,节点的坐标向量 *X*。类 NodeM 如图 2 示。

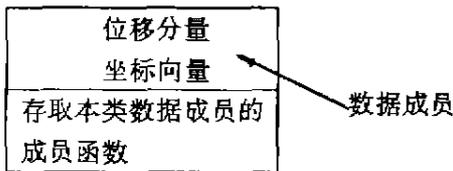


图 2 类 NodeM

其中类 Vector 和即将用到的类 InVector、Matrix 是流形方法的矩阵代数类,用于处理流形方法中的位移矢量、节点坐标向量、单元物理覆盖编号、刚度矩阵等,在文献[3,4]中有详细的定义,这里就不作具体说明。

1.2 物理覆盖节点类

物理覆盖是不连续缝对数学覆盖的再剖分,它是由计算的边界、材料分区及裂缝所组成的网格。定义在各物理覆盖上的位移函数常常被表示为级数形式,级数的每项系数 d_i 是流形方法里需求解的物理覆盖各自由度的位移

$$\begin{cases} u_i(x, y) \\ v_j(x, y) \end{cases} = \sum_{i=1}^m \begin{bmatrix} f_{\sigma}(x, y) & 0 \\ 0 & f_{\sigma}(x, y) \end{bmatrix} \begin{cases} d_{i,2i-1} \\ d_{i,2i} \end{cases} \quad (1)$$

因此物理覆盖类 NodeU 的数据成员有:物理覆盖位移自由度数 *n-dof*;覆盖各自由度的位移向量 *u*;同一数学覆盖的下一个物理覆盖 *next*。类 NodeU 如图 3 示。

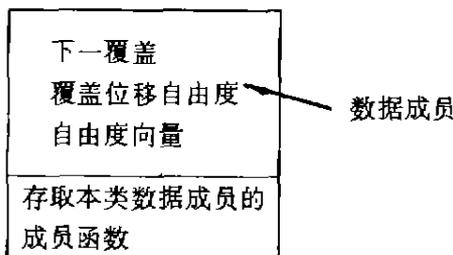


图 3 类 NodeU

1.3 流形单元类

流形单元是物理覆盖的公共区或交集。流形单元的“单元”和“节点”是相对于有限元对应体的一种扩展,用新的流形方法的节点和单元,裂

缝可以张开和滑移,块体可以移离错位,而材料体的连续区域仍可连接在一起。

数值流形方法里流形单元的积分区域是由许多直线段连接而成的多边形,流形单元的节点号(物理覆盖号)是包含此单元区域的 3 个物理覆盖的覆盖号,单元的加权系数由对应数学覆盖的节点坐标来定义。

故流形单元类 Element 的数据成员有单元材料 *mat_num*,单元积分区域节点总数 *en_count*,流形单元的数学覆盖向量 *menm*,物理覆盖向量 *uenm*,积分区域节点向量 *inenm*。流形单元类 Element 如图 4 示。

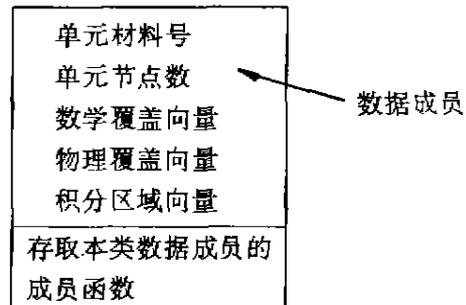


图 4 类 Element

1.4 裂缝节点类

对于存在裂纹的物体,在承受外载时,裂纹达到起裂条件开始扩展。在裂纹扩展的过程中,每隔一小段位移记录下 1 个裂纹节点,就可以形成一颗裂纹节点树。裂纹节点的数据信息包括:裂纹所处的单元号 *ele_num*;节点坐标向量 *x*;总的节点编号 *nod_num*。将裂纹节点的信息组合在一起形成一个裂纹节点类 Crack 如图 5 示。

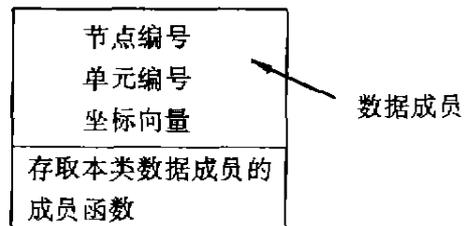


图 5 类 Crack

1.5 数据类的管理

对上述流形方法的数据类 NodeM、NodeU、Element、Crack 的对象管理采用文献[3,4]的类 Tree 将对象组织成树状结构。例如树 CrackTree 用于管理物体的所有裂纹节点对象,用 CrackTree 表示一个实例 *ckt*,则 *ckt(i)* 表示第 *i* 个裂纹节点。类 CrackTree 的定义如下:

```

class CrackTree:public Tree<Crack>
{
public:
    void cal_mess(NodeMTree &nm, ElementTree &elt);
};
    
```

```

// 计算裂纹节点的信息
void recons(NodeMTree &nmt, ElementTree &elt);
// 重新处理裂纹树
void inte_area(NodeMTree &nmt, ElementTree &elt);
// 裂纹贯穿的流形单元积分区域的记录;
void uniform(NodeMTree &nmt, NodeUTree &nur, ElementTree &elt); // 物理覆盖的记录;

```

2 物理覆盖的自动剖分

图 6 所示的物体用 8 个三角形有限单元作为有限数学覆盖。裂纹尖端位于①处,此时流形单元为 8 个;当物体承受外载达到起裂时沿②③④⑤⑥开始扩展,流形单元依次为 9、10、11、12 个。被裂缝分割的任一单元有完全不同的节点或物理覆盖码,流形单元物理覆盖的编码和积分区域节点号的编码可以在裂缝扩展的过程中手工输入,但是这样既容易造成错误,也比较费时费事。因此,完成物理覆盖系统的自动剖分,对数值流形方法的应用是一件刻不容缓的事。

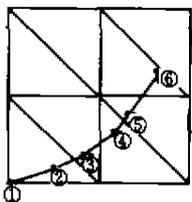


图 6 扩展中的裂纹

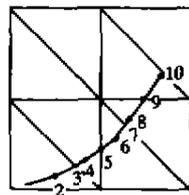


图 7 重构后的裂纹树

2.1 裂纹树的重构

在裂纹扩展过程记录下的裂纹节点可能在单元的边界上、单元内部、单元的节点上等,当裂纹贯穿原来的单元后,流形单元的积分区域和物理覆盖编码就必须重新划分。因此,对记录下的裂纹树要进行重构,把裂纹与单元边界的交点作为新的裂纹节点加入裂纹树中。

重构裂纹树的算法描述如下:

```

step1 计算裂纹树 ckt 的各节点所在的单元号
step2 向数学覆盖树里加入一个新的节点
step3 对 i = 1 到 ckt.g.num1(裂纹树的总节点数)做
step4 ~ step9
step4 判断 ckt(i+1)是否被重构过,若是则跳到 step3
step5 判断 ckt(i)或 ckt(i+1)是否在所属单元的边界中,若是则跳到 step9
step6 判断 ckt(i)或 ckt(i+1)是否在同一个单元,若是则跳到 step9
step7 判断 ckt(i)或 ckt(i+1)与单元边界的交点 Pi,i+1

```

```

step8 将 Pi,i+1作为新节点加入到裂纹树里

```

```

step9 向数学覆盖树里加入一个新的节点

```

经过重构后的裂纹如图 7 示。一般说来,当裂纹节点跨越单元时,裂纹树要重构一次。

2.2 积分区域的记录

经过重构后的裂纹树,新自动生成的节点编号如图 8 示。对被裂纹划分的流形单元,从原始单元的 1 个节点出发逆时针寻找并记录该单元的积分区域的节点号,然后向单元树里加入 1 个单元对象。例如原始单元 1,2,4 被切割后的两个流形单元的区域分别为 1, 10, 11, 2 和 4, 11, 10, 1。

2.3 单元物理覆盖的自动剖分

物理覆盖的区域是包含在数学覆盖中的材料,如果材料边界、块体边界或裂缝把数学覆盖划分成完全的隔离区域,每一区就是物理覆盖。因此,物理覆盖是数学覆盖的再剖分。

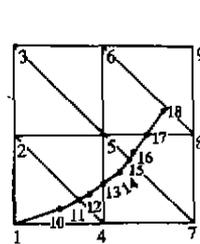


图 8 裂纹树的整体节点编号

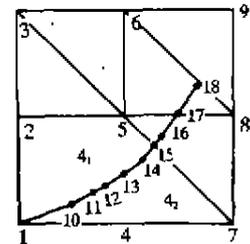


图 9 节点 4 的物理覆盖系统

在图 9 中,节点 4 的数学覆盖 V₄ 是由多边形 7, 5, 2, 1, 4 确定的面积,当裂缝穿透数学覆盖 V₄ 时, V₄ 就被划分成两个物理覆盖 4₁, 4₂. 则物理覆盖 4₁ 是多边形 5, 2, 1, 11, 12, 13 确定的面积, 4₂ 是 1, 4, 7, 15, 14, 13, 12, 11, 10 确定的面积。图 8 中流形单元 4, 11, 10, 1 包含于物理覆盖 4₁ 的区域,故其对应物理覆盖号是 4₁;流形单元 2, 1, 10, 11 包含于物理覆盖 4₂ 的区域,故其对应物理覆盖号是 4₂。

因此对整个物理覆盖系统,先判断每一节点数学覆盖被裂缝分割的物理覆盖个数,进行计算机自动编码,并记下对应物理覆盖的区域,再从流形单元的原始数学覆盖出发去求流形单元与物理覆盖的交集,就可完全自动地找出各流形单元的物理覆盖号。

物理覆盖的自动剖分生成算法可描述如下:

```

step1 对 i = 1 到 num.g.num1(数学覆盖树的总节点数)做
step2 ~ step3
step2 判断数学覆盖是否被裂缝贯穿,若没有则跳转到

```

- step1
- step3 增加新的物理覆盖节点到物理覆盖树
- step4 对 $n=1$ 到 $elc.g.num1$ (单元树的总单元数) 做 step5 ~ step6
- step5 对 $k=1$ 到 enc (单元节点总数) 做 step6
- step6 从单元的积分区域节点号和物理覆盖树里寻找对应的物理覆盖号, 并记入单元的物理覆盖编码

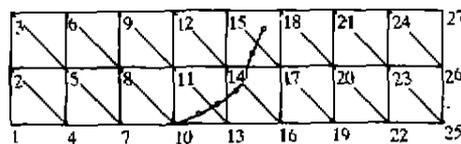


图 10 悬臂梁

3 算例

用三角形有限元网格作为数学覆盖网格的悬臂梁, 其原始单元和节点编码如图 10 示。裂纹尖端出现在节点 10 处, 受载后, 裂纹不断扩展到达节点 38。图 11 中的裂纹是经过重构后的裂纹树。用本算法生成的被裂缝贯穿的流形单元的各种编码见表 2。需要注意的是数学覆盖、积分区域和物理覆盖是三套独立的编码, 不能将它们混淆在一起。

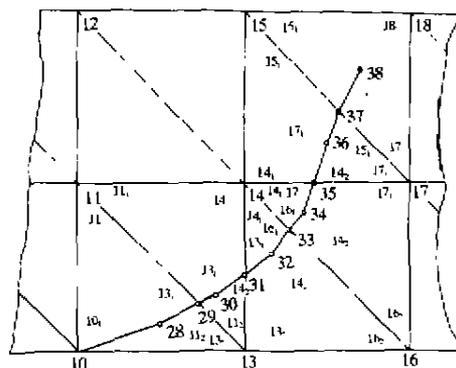


图 11 悬臂梁的物理覆盖系统

比较表 2 中的物理覆盖示意编码和程序生成的计算编码可以看出本算法是正确有效的, 能够用于连续体的物理覆盖系统网络的自动生成。

表 2 悬臂梁流形单元的剖分编码

原始单元	积分区域(程序生成)	流形单元	物理覆盖示意编码	物理覆盖计算编码(程序生成)
10, 13, 11	10, 28, 29, 11, 13, 29, 28, 10	$10_1, 13_1, 11_1, 10_2, 13_2, 11_2$	$10_1, 13_1, 11_1, 10_2, 13_2, 11_2$	10, 13, 11, 28, 30, 29
11, 13, 14	11, 29, 30, 31, 14, 13, 31, 30, 29	$11_1, 13_1, 14_1, 11_2, 13_2, 14_2$	$11_1, 13_1, 14_1, 11_2, 13_2, 14_2$	11, 13, 14, 29, 30, 31
13, 16, 14	14, 31, 32, 33, 16, 33, 32, 31, 13	$13_1, 16_1, 14_1, 13_2, 16_2, 14_2$	$13_1, 16_1, 14_1, 13_2, 16_2, 14_2$	13, 16, 14, 30, 32, 31
14, 16, 17	14, 33, 34, 35, 17, 35, 34, 33, 16	$14_1, 16_1, 17_1, 14_2, 16_2, 17_2$	$14_1, 16_1, 17_1, 14_2, 16_2, 17_2$	14, 16, 17, 31, 32, 17
14, 17, 15	14, 35, 36, 37, 15, 17, 37, 36, 35	$14_1, 17_1, 15_1, 14_2, 17_2, 15_2$	$14_1, 17_1, 15_1, 14_2, 17_2, 15_2$	14, 17, 15, 31, 17, 15

4 结语

根据数值流形方法的基本理论, 引入面向对象的设计思想, 实现了数值流形方法物理覆盖系统的自动剖分算法, 对于数值流形方法的应用具有实际意义。文中建立的物理覆盖系统的数据类具有通用性, 可以用于面向对象的数值流形方法的程序设计中。

参 考 文 献

[1] 石根华著. 数值流形方法与非连续变形分析[M]. 北京: 清华大学出版社, 1997.
 [2] 王水林. 流形元方法在模拟裂纹扩展中的应用[J]. 岩石力学与工程学报, 1997, 16(5): 405~410.
 [3] SCHOLZ S P. Elements of an Object-Oriented FEM++ Program

in C++[J]. Comput. Struct, 1992, 43(3): 517~529.
 [4] 周本宽. 面向对象的有限元程序的类的设计[J]. 计算结构力学及其应用, 1996, 13: 269~278.
 [5] FORDE B W R. Objected-Oriented Finite Element Analysis[J]. Comput. Struct, 1990, 35(3): 355~374.
 [6] MILLER G R. An Objected-Oriented Approach to Structural Analysis and Design[J]. Comput. Struct, 1991, 40(1): 75~82.
 [7] 王芝银. 岩石大变形分析的流形方法[J]. 岩石力学与工程学报, 1997, 16(5): 399~404.
 [8] 裴觉民. 数值流形方法与非连续变形分析[J]. 岩石力学与工程学报, 1997, 16: 279~292.
 [9] 王芝银, 李云鹏. 数值流形方法中的几点改进[J]. 岩土工程学报, 1998, 20: 33~36.

(下转 44 页)

