

MatDEM 中的单元和组

1 MatDEM 中的单元类型

1.1 活动单元、固定单元、虚单元

MatDEM 中主要有两种基本的单元类型，即活动单元（模型单元）和固定单元（也称墙单元），如图 2.3-1 所示。

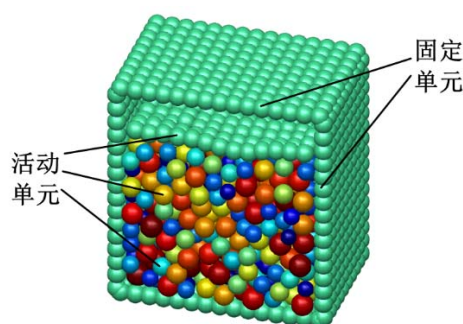


图 2.3-1 MatDEM 中的两种基本单元类型

在数值模拟时，软件计算活动单元受力，并使之运动（除非单元被锁定自由度），单元编号在 1 到 `mNum` 之间（见 2.3.3 节）；而对于固定单元，不直接计算单元的受力，且单元保持不动（除非通过命令移动它），单元编号在 `mNum+1` 至 `aNum-1` 之间。其中，固定单元又分为两种：第一种固定单元是程序预设的边界单元，如模拟箱（`obj_Box`）的 6 个边界面上的单元，它们是特殊的固定单元；第二种固定单元是普通的固定单元（`wall`），用于构建模型中间不动的物体。

在 MatDEM 后处理模块生成的图片中，固定单元默认以绿色来显示。但需要注意的是，绿色的单元不一定是固定单元。例如，在执行命令 `d.show('aR')` 时，由于压力板单元的半径接近单元平均半径（`B.ballR`），因此常显示为绿色，但压力板单元均为活动单元。由于软件不计算固定单元的受力和运动，当向模型中添加大量固定单元时，基本不会增加数值模拟的计算量。在使用函数 `d.addElement` 向模型中添加单元时，可通过第三个输入参数声明所添加单元的类型，其取值为 `'model'`（活动单元）或 `'wall'`（固定单元），默认为 `'model'`。在将单元加入模型后，可通过函数 `d.defineModelElement` 和 `d.defineWallElement` 重新定义活动单元和固定单元。'

模型中还有一个非常特殊的虚单元，其编号为 `aNum`。它并不是实际单元，与其余的实际单元之间没有力的作用，也不会运行后处理命令时显示出来，只起标记作用。MatDEM 的高效率离散元数值模拟基于矩阵计算，由于每个单元的邻居单元数不一致，因此为了构建邻居矩阵，需要将邻居矩阵中长短不一的位置用编号 `aNum` 来填充，即引入了虚单元，所有前缀为 `a` 的参量数组的最后一个元素都属于虚单元。MatDEM 在计算某一单元的受力时，先计算该单元与其所有邻居单元间的作用力，然后再判别邻居单元的编号，如果为 `aNum`，则把它与该单元之间的作用力赋为零，以准确地计算单元与其邻居单元间的作用力。虚单元坐标始终与第 `aNum-1` 号单元重合，同时半径为其 $1/4$ 。

1.2 锁定单元自由度

活动单元的自由度可以通过 `build` 类中 `addFixId` 函数锁定。通过锁定单元自由度可以限制物体的运动方向，命令为 `d.addFixId(direction, gId)`，其中 `direction` 为需要锁定自由度的方向，可取 `'X'`、`'Y'` 或 `'Z'`，`gId` 为单元编号数组。或者，修改对象 `d.mo`（通过 `model` 类创建）中

的 `FixXId`、`FixYId` 和 `FixZId` 矩阵也可以直接锁定单元某方向自由度。例如，在示例 `BoxPile` 中，将桩体沿 `X` 方向的自由度锁定，从而使得二维平面内的桩只能沿 `Z` 方向上下运动。此外，在通过压力板对模型施加定向压力时，也需通过命令 `B.setPlatenFixId()` 锁定除压力板法线方向以外的自由度，以免压力过大时压力板从样品边缘滑落。

当活动单元的全部自由度均被锁定时，其效果与固定单元一致，但仍属于活动单元。锁定单元自由度的本质是通过 `d.mo` 中的 `FixXId`、`FixYId` 和 `FixZId` 来记录单元锁定信息，在平衡迭代时，这些单元不会在被锁定自由度的方向上产生位移。但是，仍可通过函数 `d.moveGroup` 或修改 `d.mo` 中的单元坐标来直接移动它们。可以通过函数 `d.removeFixId` 解锁单元的自由度。

1.3 单元编号的规则

在 `MatDEM` 中，单元总数为 `aNum`，记录在 `d.aNum` 和 `d.mo.aNum` 中。每个单元都有唯一的编号，编号为 1 至 `aNum`。其中活动单元的个数 `mNum`，记录在 `d.mNum` 和 `d.mo.mNum` 中，编号为 1 至 `mNum`；而编号为 `mNum+1` 至 `aNum-1` 的单元为固定单元；编号为 `aNum` 的单元为虚单元。因此，在 `MatDEM` 中，活动单元的编号永远比固定单元的小。若将活动单元通过函数 `d.defineWallElement` 转化为墙单元，如把某压力板的单元重新声明为墙单元，则其编号也会发生变化，反之亦然。编号变化及其带来的一系列数据调整均由 `MatDEM` 自动完成。

2 MatDEM 的数据结构

2.1 单元属性数组

单元的各类属性信息主要记录在 `d` (`build` 类的对象) 和 `d.mo` (`model` 类的对象) 中的属性数组中。其中，`d` 中的属性数组记录了单元的初始状态，而 `MatDEM` 的核心计算模块是 `d.mo`，其记录了模型单元当前的状态。

`d.mo` 中的很多基本属性是活动单元和固定单元所共有的，例如三维坐标 (`aX`、`aY`、`aZ`)，单元半径 (`aR`)，微观力学性质参数 (法向劲度系数 `aKN`、切向劲度系数 `aKS`、临界断裂力 `aBF`、初始抗剪强度 `aFS0`、粒间摩擦系数 `aMUp`)，等。这些属性数组名均以字母 `a` 开头 (`all`)，数组的长度为 `aNum`，数组行号对应单元的编号，数组值记录了对应单元的属性值，如 `d.mo.aX(1)` 记录了第一个单元的 `x` 坐标。

由于固定单元不会运动，因此在运动相关的属性数组中只需记录活动单元的值，此类数组包括：单元的质量 (`mM`)、阻尼 (`mVis`)、三个方向的速度 (`mVX`、`mVY`、`mVZ`)、加速度 (`mAX`、`mAY`、`mAZ`)、阻尼力 (`mVFX`、`mVFY`、`mVFZ`)、体力 (`mGX`、`mGY`、`mGZ`，其中 `mGZ` 相当于重力)。这些属性数组名均以字母 `m` 开头 (`model`)，数组的长度为 `mNum`，以此表明它们是活动单元所独有的参数。关于这些参数，详见帮助中的 `build` 属性和 `model` 属性表。

2.2 邻居矩阵和连接信息矩阵

单元的邻居矩阵 (`d.mo.nBall`) 是 `MatDEM` 中的一个重要概念。邻居矩阵共有 `mNum` 行，且与活动单元一一对应，逐行记录了每个活动单元的邻近单元编号。邻居矩阵的列数通常为几列至几十列。图 2.4-1 为一个二维模型的邻居矩阵，为 757 行 10 列的矩阵。邻居矩阵的列数由最多邻居单元的行所决定，三维模型的单元会有更多的邻居单元，其邻居矩阵通常为数十列。当大直径单元附近有大量小直径单元时，其邻居单元数会显著增加。`nBall` 矩阵的列数通常随着最大最小粒径比的增加而增加。为了保证合适的计算量，建议 `MatDEM` 中基本单元粒径比不要超过 5，并可以使用 `clump` 来构建较大的颗粒 (详见 2.6 节)。

由于每个单元 (每行) 的邻居单元数不一致，为了构建列长度相同的邻居矩阵，需要将邻居矩阵中长短不一的位置用编号 `aNum` (虚单元) 来填充。例如，在下图中，`aNum` 为 898，数据表明：第一个单元有 9 个邻居单元 (除去 898 单元)，同理第二个单元有 7 个邻居单元。本质上，`nBall` 矩阵记录了单元间的连接关系，即哪些单元间构成连接。

	d		d.mo		d.mo.nBall					
	1	2	3	4	5	6	7	8	9	10
1	2	26	27	626	627	658	692	693	724	898
2	1	3	27	28	692	693	694	898	898	898
3	2	4	28	29	694	695	696	898	898	898
4	3	5	29	30	695	696	697	898	898	898
5	4	6	30	31	696	697	698	898	898	898
6	5	7	31	32	697	698	699	898	898	898
7	6	8	31	32	33	698	699	700	898	898
8	7	9	32	33	700	701	702	898	898	898
9	8	10	33	34	35	701	702	703	898	898
10	9	35	36	703	704	898	898	898	898	898
11	12	36	37	704	705	706	898	898	898	898
12	11	13	37	705	706	707	898	898	898	898
13	12	14	38	39	706	707	708	709	898	898
14	13	15	39	40	708	709	710	898	898	898
15	14	16	40	709	710	711	898	898	898	898

图 2.4-1 邻居矩阵 d.mo.nBall，数值代表单元编号

MatDEM 的迭代计算基于 nBall，通过 nBall 可以得到一系列与 nBall 相同大小，记录单元与其邻居单元间连接关系和属性值的矩阵，称为连接信息矩阵，包括过滤器矩阵和属性矩阵。

过滤器矩阵为布尔矩阵，以 d.mo.cFilter 为例（图 2.4-2），cFilter 与 nBall 矩阵中的单元编号一一对应，记录了单元与其邻居间是否处于压缩状态（当其值为 1 时则处于压缩状态）。结合 nBall 和 cFilter 矩阵可以得到所有单元连接的压缩状态。如 nBall 矩阵的第一行第一列值为 2，而 cFilter 相应位置的值为 1，说明单元 1 和单元 2 的连接目前处于压缩状态。同样地，利用 bFilter 矩阵可以得到单元间是否胶结，利用 tFilter 矩阵可以得到单元间是否处于张拉状态。这些过滤器矩阵记录了单元间连接的状态。

	d		d.mo		d.mo.nBall		d.mo.cFilter			
	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	0	0	1	0	0	0
2	1	1	0	0	0	1	0	0	0	0
3	1	1	1	0	0	1	0	0	0	0
4	1	1	0	1	0	1	0	0	0	0
5	1	1	1	0	0	1	0	0	0	0
6	1	1	0	0	0	1	0	0	0	0
7	1	1	0	1	0	0	1	1	0	0
8	1	1	0	0	0	1	0	0	0	0
9	1	0	0	1	0	0	1	0	0	0
10	0	1	0	1	1	0	0	0	0	0
11	1	1	1	0	1	0	0	0	0	0
12	1	1	1	0	1	0	0	0	0	0
13	1	0	0	1	0	1	1	0	0	0
14	0	1	1	0	0	1	0	0	0	0
15	1	1	1	0	1	0	0	0	0	0

图 2.4-2 单元连接压缩状态过滤器矩阵 cFilter，1 代表连接处于压缩状态

属性矩阵通常为数值矩阵（双精度浮点数），以 d.mo.nFnX 为例（图 2.4-3），nFnX 与 nBall 矩阵中的单元编号一一对应，记录了单元间法向弹簧力在 x 方向上的分量。同样的，结合 nBall 和 nFnX 矩阵可以得到所有单元连接的法向弹簧力在 x 方向上的分量。如 nBall 矩阵的第一行第一列值为 2，而 nFnX 相应位置的值为 -0.0145，说明单元 1 受单元 2 法向弹簧沿 x 负方向 0.0145 牛的力。同样地，利用 nFsX 矩阵可以得到单元间切向弹簧力在 x 方向上的分量；nFsX 与 nFnX 相加得到 nFX 矩阵，即单元受其邻居总作用力在 x 方向上的分量；将 nFX 在水平方向上求和后，则可得到每个单元受邻居单元的合力在 x 方向上的分量。

这类属性矩阵包括：连接的残余强度系数（nBondRate）、连接的刚度矩阵（nKNe、nKSe、nIKN、nIKS）、法向弹簧力的三个分量（nFnX、nFnY、nFnZ）、切向弹簧力的三个分量（nFsX、nFsY、nFsZ）、clump 连接的初始重叠量（nClump），等。这些属性矩阵记录了单元间连接的属性。以 nFnX 为例，其命名法则为：neighboring Force of normal spring in X direction。

	d	d.mo	d.mo.nBall	d.mo.nFnX						
	1	2	3	4	5	6	7	8	9	10
1	-0.0145	0.0095	-0.0093	0.0106	0	0	0.0039	0	0	0
2	0.0145	-0.0155	0	0	0	9.6187e-04	0	0	0	0
3	0.0155	-0.0174	0.0101	0	0	-0.0083	0	0	0	0
4	0.0174	-1.8066e-04	0	-0.0165	0	-8.2588e-04	0	0	0	0
5	1.8066e-04	-9.1296e-04	5.1487e-04	0	0	2.4837e-04	0	0	0	0
6	9.1296e-04	-9.5839e-04	0	0	0	5.1437e-05	0	0	0	0
7	9.5839e-04	-0.0019	0	-0.0043	0	0	0.0086	-0.0033	0	0
8	0.0019	-0.0019	0	0	0	-4.7287e-05	0	0	0	0
9	0.0019	0	0	-0.0024	0	0	5.0151e-04	0	0	0
10	0	0.0015	0	5.4046e-04	-0.0021	0	0	0	0	0
11	-0.0063	0.0098	-0.0037	0	2.4243e-04	0	0	0	0	0
12	0.0063	-0.0071	4.5696e-04	0	2.3333e-04	0	0	0	0	0
13	0.0071	0	0	-0.0141	0	0.0078	-5.2907e-04	0	0	0
14	0	-0.0041	0.0049	0	0	-8.3566e-04	0	0	0	0
15	0.0041	-0.0109	0.0058	0	0.0011	0	0	0	0	0

图 2.4-3 单元间法向弹簧力在 x 方向上的分量 nFnX，数值代表受力值

2.3 组的数据结构和操作

组的数据结构

在 MatDEM 中，可以将任意一组编号的单元定义成组，然后再对组进行各类操作。例如，在初始化建模时，自动生成的边界组（如 lefB），以及压力板组（如 topPlaten）。如图 2.4-4 所示，这些组的单元编号记录在 d.GROUP 中，如 d.GROUP.lefB 记录了左边界的单元编号，d.GROUP.sample 记录了模型箱中间样品的单元编号，等。

命令编辑器					数据查看器				
	d	d.GROUP							
	1	2	3	4					
1	lefB	240×1 double	1170	1409					
2	rigB	240×1 double	1410	1649					
3	froB	240×1 double	1650	1889					
4	bacB	240×1 double	1890	2129					
5	botB	225×1 double	2130	2354					
6	topB	225×1 double	2355	2579					
7	lefPlaten	0×1 double							
8	rigPlaten	0×1 double							
9	froPlaten	0×1 double							
10	bacPlaten	0×1 double							
11	botPlaten	0×1 double							
12	topPlaten	169×1 double	1001	1169					
13	sample	1000×1 double	1	1000					
14	groupId	2580×1 double	-6	10					
15	groupProtect	6×1 cell							
16	groupMat	1×1 struct							

图 2.4-4 MatDEM 中的组

d.GROUP 记录了模型中所有的组和组的信息，主要包括三类：

- （1）边界组，即由建模器产生的六个默认边界（左边界 lefB、右边界 rigB、前边界 froB、后边界 bacB、底边界 botB、顶边界 topB），边界组均为固定单元；
- （2）普通组，包括①由系统自动建立的六块压力板，当压力板不存在时，其相应的矩阵为空，即[]；②初始建模时，模拟箱内部的样品组 sample；③使用函数 d.addGroup 可自定义新的组，它们都会被自动记录在 d.GROUP 中。注意：自定义的组名不能以'group'开头。
- （3）d.GROUP 中以 group 开头的矩阵记录了组和组单元的信息，包括：①groupId：记录每个单元所属组的编号，模型中每个单元都有一个组编号，数组长度为 aNum。MatDEM 的单元组号准则如下：未进行分组的单元组号为 0；左、右、前、后、下、上边界单元的组号为 -1 至 -6；左、右、前、后、下、上压力板单元的组号为 1 至 6；sample 的组号为 10。在使用 d.addElement(matId,addObj)命令增加单元时，如果结构体 addObj 中存在 addObj.groupId 数组时，则新单元的组号由结构体里的 groupId 定义；如不存在 groupId，则组号为 0。groupId 在建立 clump 的过程中有重要作用，具体请见 2.6.2 节。②groupProtect：记录受保护组的组

名，其单元不会被 `d.delElement` 删除。`groupProtect` 默认会记录 6 个边界组名，以防止切割模型时将边界单元删除，并造成活动单元飞出模型箱区域。也可以将普通组的组名加入到 `groupProtect` 中，防止其被误删。③`groupMat`：记录模型中所有单元的材料编号。`groupMat` 为结构体，通过 `d.setGroupMat` 命令来设置，并通过 `d.groupMat2Model` 命令将组的材料赋到模型中。

组的操作函数

MatDEM 的 `build` 类提供了一系列组的操作函数。表 2.4-1 给出了相关函数，具体用法请参见本书附录或 MatDEM 根目录下的帮助文档。

表 2.4-1 MatDEM 中的部分组操作函数

函数名	功能
<code>addGroup</code>	在当前模型中定义一个新组，
<code>breakGroup</code>	断开指定组内连接或两个组间的连接
<code>breakGroupOuter</code>	断开指定组向外的连接
<code>connectGroup</code>	胶结指定组内连接或两个组间的连接
<code>connectGroupOuter</code>	胶结指定组向外的连接
<code>delGroup</code>	删除指定名称的组（并不会删除单元）
<code>removeGroupForce</code>	忽略两个组间的所有作用力
<code>rotateGroup</code>	旋转指定组的单元
<code>minusGroup</code>	将两个相互重叠的组相减
<code>moveGroup</code>	强制移动某组，包括固定单元和锁定自由度的单元
<code>protectGroup</code>	将组声明为受保护，并记录于 <code>groupProtect</code> 中

使用 `addGroup(gName,gId,varargin)` 可以将指定的单元定义一个组，其中 `gName` 为组名；`gId` 为单元的编号数组；`varargin` 为可选参数，可以输入组的材料号（即 `d.Mats` 中的序号）。组是特定单元编号的集合，当新增或删除某个组时，模型中的单元数并不会增减。

在建立复杂模型的过程中，还可能会用到组相减函数 `minusGroup`，该函数的第一个输入参数是要被移除单元的组（被减），第二个是保持完整的组（减），而第三个输入参数则是完整组的半径比率，具体见 6.1.2 节。另外，可以通过函数 `group2Obj` 将组转化成结构体，供其它模型使用，具体见 3.2.1 节；也可以利用过滤器筛选出特定的单元，然后快速建组，具体见 3.2.2 节。

3 单元的接触模型

MatDEM 默认采用线弹性接触模型，但仍提供了定义法向接触力的接口，即类 `model` 的属性 `d.mo.FnCommand`。使用者可以自定义 `FnCommand` 的内容来实现其他非线性接触模型的定义。根据使用者的需求，MatDEM 将开放接触模型的接口，使用者可自由定义各类接触模型，实现复杂材料的定义。

3.1 线弹性模型

在 1.2.1 节中，我们已经介绍了线弹性模型的基本原理。在程序中，当接触模型为线弹性时，`FnCommand` 参数的取值如下：

```
FnCommand='nFN0=obj.nKNe.*nIJXn;';
```

3.2 赫兹接触模型

当颗粒表面光滑无黏连，接触面与总表面积相比极小，接触力垂直于接触面，且仅有弹性形变发生时，单元之间的法向接触力可采用赫兹模型计算。如图 2.5-1 所示，半径分别为 R_1 、 R_2 的两单元发生弹性接触，法向重叠量 A 满足：

$$A = R_1 + R_2 - |\mathbf{r}_1 - \mathbf{r}_2| > 0 \quad (2.5-1)$$

式中， \mathbf{r}_1 、 \mathbf{r}_2 分别为两单元的位置矢量。接触面为圆形，其半径 a 满足：

$$a = \sqrt{A \frac{R_1 R_2}{R_1 + R_2}} \quad (2.5-2)$$

在离散元法中单元泊松比 μ 为 0，且单元弹性模量 E 与法向刚度 K_n 之间存在如下关系：

$$E = \frac{K_n}{\pi R} \quad (2.5-3)$$

故法向力 F_n 为

$$F_n = \frac{4(R_1 + R_2)k_{n1}k_{n2}}{3\pi R_1 R_2 (k_{n1} R_2 + K_{n2} R_1)} a^3 \quad (2.5-4)$$

关于赫兹接触模型，可参考颗粒介质力学相关专著。在程序中，通过修改 `d.mo.FnCommand` 字符串来定义赫兹模型，如在示例文件 `3AxialNew2` 中，将 `FnCommand` 赋值为（MatDEM1.32 的编译器暂不支持续行功能，因此以下代码实际应写在同一行中）：

```
d.mo.FnCommand='nFN1=obj.nKNe.*nIJXn;
nR=obj.aR(1:m_Num)*nRow;
nJR=obj.aR(obj.nBall);
Req=nR.*nJR./(nR+nJR);
nE=obj.aKN(1:m_Num)*nRow./(pi*nR);
nJE=obj.aKN(obj.nBall)./(pi*nJR);
Eeq=nE.*nJE./(nE+nJE);
nFN2=-4/3*Eeq.*Req.^(1/2).*abs(nIJXn).^(3/2);
f=nIJXn<0;
nFN0=nFN1.*(~f)+nFN2.*f;';
```

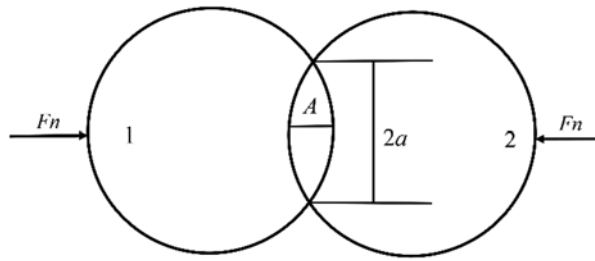


图 2.5-1 赫兹接触模型示意图

4 弹性 clump 团簇

4.1 弹性 clump 的原理

MatDEM 的基本单元均为小球，可以将若干小球相互交叠构成 **clump**（团簇），以实现非球形颗粒和物体的建模。如图 2.6-1 所示：在 **clump** 模型中，两个单元相互重叠，且重

叠量为 l_0 ，单元直径为 d ，设定其平衡距离为 $d-l_0$ ，两单元球心之间的距离为 r ，则单元间相对位移由以下公式计算得到

$$X_n = r - (d - l_0) \quad (2.6-1)$$

根据此公式，在图 2.6-1a 的状态时，两单元的相对位移 X_n 为零，处于平衡状态。而图 2.6-1a 中，双向箭头处可视为两单元的接触表面。在图 2.6-1b 中，当两单元间距离增加后， X_n 增加，单元间产生拉力。同样地，当相对位移达到断裂位移 X_b 时，连接断裂（图 2.6-1c）。当两单元相互挤压时，单元间产生压力（图 2.6-1d）。事实上，团簇模型将两单元间的平衡距离缩小，在计算相对位移时减去其初始的重叠量。通过这种方法，可以实现较为复杂模型的建立。如图 2.6-1e，四个单元相互重叠，令其受力平衡，其实际接触如图 2.6-1f 所示。通过重叠单元，可以构建表面较为光滑的团簇模型。

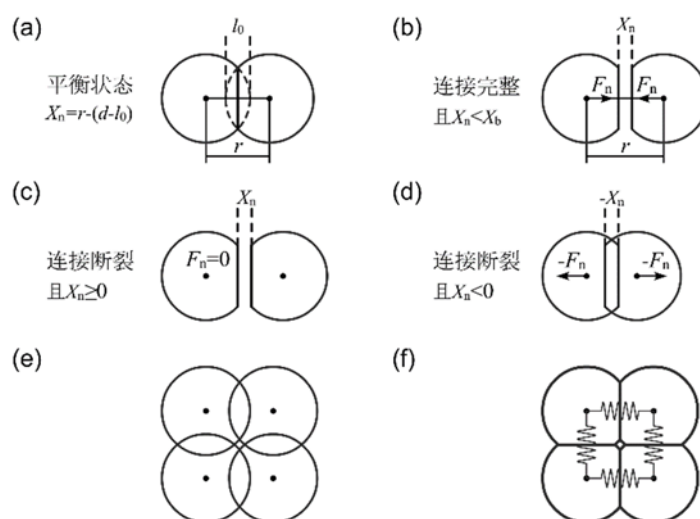


图 2.6-1 团簇 clump 示意图

当一个连接被声明为 **clump** 时，会将连接对应的单元重叠量设为初始重叠量（即此时弹簧变形量为 0，颗粒间无相互作用），并记录在 **d.mo.nClump** 矩阵中。在计算两单元受力时，会将单元间的实际重叠量减去初始重叠量，**clump** 单元之间的相对位置不是固定不变的，仍然存在受力和变形。因此，MatDEM 中的 **clump** 团簇是可变形的。

初始重叠量仅针对两个特定单元的连接，当连接断开，两单元与其他单元接触时，初始重叠量将不复存在，并错误地导致模型总体积增加，因此需要保证 **clump** 连接不可断裂。在每次平衡迭代时，若 **nClump** 矩阵中某个值不为零，软件会自动地把 **bFilter** 中对应位置的值设置为 1，并使其不可断裂（除非应变非常大，见后）。但是，邻居矩阵 **nBall** 仅包含一定距离内的邻居单元编号（由 **d.mo.dSide** 决定），这个距离默认约为 0.4 倍的平均单元半径。如果 **clump** 连接的两个单元的距离超出邻居范围，**clump** 连接也会断裂。大多数岩土体材料的抗拉应变无法达到这种程度。如果需要模拟橡皮筋等抗拉应变特别大的材料，可以让单元间的初始重叠量大一些，例如在 **BoxSlopeNet2** 中，通过以下命令来构建网：

```
netObj=ms.denseModel(0.8,@ms.makeNet,B.sampleL*2,B.sampleL/3*2,cellW,cellH,B.ballR);
```

将第一个输入参数 0.8 (**Rrate**) 改小一些，则可提高 **clump** 材料的抗拉应变。通常 **Rrate** 应大于 0.5，以避免相隔较远的 **clump** 单元间产生作用。同时，较小的 **Rrate**（如 0.1）会导致大量单元拥挤在一起，并增加计算量（**nBall** 列数增加）。

此外，构成 **clump** 连接的单元也可以是分开无接触的，即超距作用。当然 **clump** 单元不能间隔太远，否则需要增大 **d.mo.dSide**（搜索邻居单元时的查找范围）。这个功能极少使用，在此不做深入介绍。

4.2 弹性 clump 的使用

在建立初始几何模型时 (**B.buildInitialModel**)，如设置 **B.isClump=1**（默认为 0），软件会通过 **B.createClump(B.distribRate)** 自动生成一系列团簇颗粒 **clump**，其分散系数 **B.distribRate** 越大，生成的 **clump** 越不规则。示例 **BoxSlope1-2** 演示了如何自动生成 **clump**。图 2.6-2 为

BoxSlope1 示例生成的 clump 堆积体，每个 clump 由 2~8 个基本单元组成。系统通过 d.GROUP.groupId 来区分不同 clump，这些自动生成的 clump 的组号均从-11 开始递减。在运行 d.setClump()命令时，组号等于-11 的单元会被视为属于同一个 clump，同样的，-12，-13 等同样组号的单元会被视为同一个 clump，并将其初始重叠量存储在 d.mo.nClump 中，从而实现自动构建 clump 颗粒。

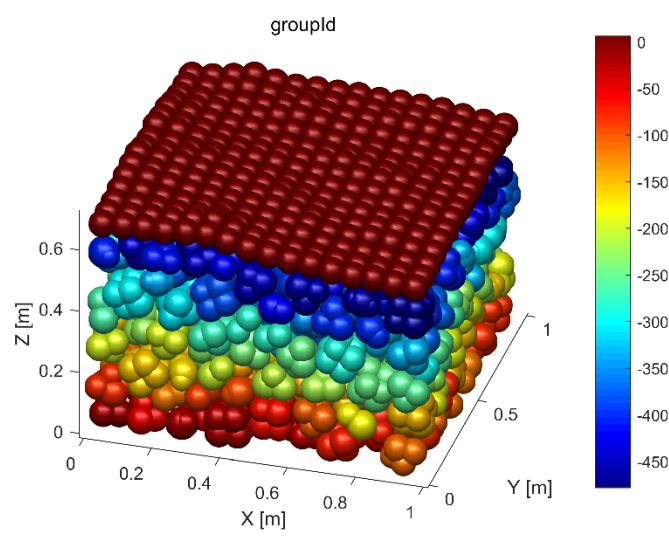


图 2.6-2 MatDEM 自动生成的团簇 clump

这些 clump 颗粒都属于 sample 组，通过命令 d.group2Obj 可以将组转成结构体，并进一步导入到新的模型中。例如，在 BoxSlope2 中，通过以下代码实现组和结构体的转化：

```
packBoxObj=d.group2Obj('sample')
...
boxObjId=d.addElement(1,packBoxObj);
```

当运行 group2Obj 命令后，将 sample 组中单元信息转成结构体 packBoxObj。从图 2.6-3a 中可以看到，结构体中包括了单元的坐标、半径和 groupId，其中，groupId 由-11 递减到 -406。图 2.6-2 中的左下角的第一个 clump 包含 8 个基本单元，其对应于 groupId 的第 1 至 8 个组号，均为-11（图 2.6-3b）。使用者可以根据实际需求，编写特定的 clump 颗粒函数，生成结构体中的单元坐标、半径和组号，并将结构体导入到模型中，生成各种形态的 clump 团簇，以及 clump 与基本单元的混合堆积体。

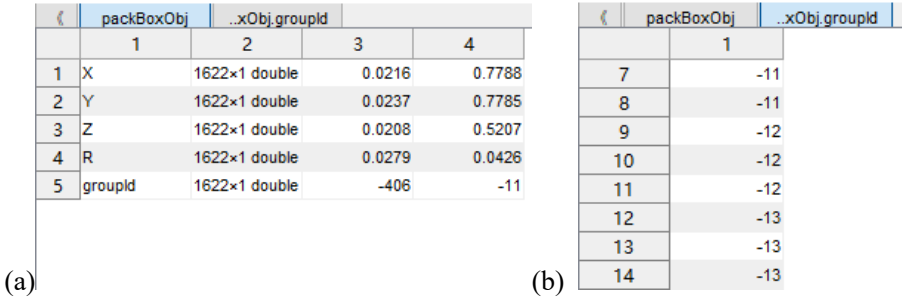


图 2.6-3 （a）clump 的结构体；（b）结构体中的 groupId 定义不同 clump 颗粒

在 MatDEM 中，clump 的实现原理基于上节所述的 nClump 矩阵。如果 nClump 矩阵中值为 0，则程序不认为相应的两单元构成 clump，连接会受拉断裂。因此，如需将没有重叠量的单元连接设为 clump，可以在 nClump 中将相应连接赋以极小的重叠量。弹性 clump 已有较为丰富的应用，也可参考 BoxSlopeNet 示例。

附录：常见问题解答

1. Q: 在建模过程中出现报错：无法从 `gpuArray` 转换为 `double`，是什么原因？

A: 出错原因为在进行非迭代计算操作时没有关闭 GPU，此时需要在出错代码前关闭 GPU，运行命令 `d.mo.setGPU('off')`。

MatDEM 二次开发基于 Matlab 语言，如出现报错提示，可将错误提示信息和“Matlab”一起进行网络搜索，可获得关于此报错的原因和解决方法。同时，需查看与报错相关的参数，分析错误原因。如无法解决，可保存相关代码和报错提示，使用 `save` 命令保存报错时的全部参数于 `.mat` 文件中，并将相关文件提交 MatDEM 在线讨论群，共同分析和解决问题。

2. Q: 怎么样可以加速模型的运动过程计算？

A: 建议适当增大默认的标准时间步 `d.mo.dT`，但时间步不要超过标准时间步的 4 倍。使用命令：

```
d.mo.dT=d.mo.dT*4;
```

3. 为什么能量曲线图中的总能量曲线（Total）不水平？

A: MatDEM 实现了能量守恒计算，在无外力做功时，总能量曲线应为水平。但是，由于时间步默认取周期的 $1/50$ ，其迭代运算时的能量计算还不够精确。当时间步取周期的 $1/200$ 时，即运行 `d.mo.dT=d.mo.dT/4` 后再进行迭代运算，能保证能量守恒计算，总能量可保持水平。

4. Q: 将结构体导入模型后，为什么结构体单元在平衡过程中发生“爆炸”？

A: 采用结构体建模，当把结构体导入模型后，如果单元间的重叠量太大，颗粒间会产生巨大的应力，并导致结构体模型爆炸和破坏。因此，在添加结构体之后，通常需要利用 `setClump` 命令将其设为团簇颗粒。

5. Q: 为什么在平衡模型后颗粒会“爆炸”和飞出边界？

A: 有如下几种可能：

(1) 在第二步材料设置步骤完成后，颗粒刚度通常会增大，并导致颗粒间作用力急剧增加，此时需要使用 `balanceBondedModel`（或 `balanceBondedModel0`）命令进行强制胶结平衡，防止颗粒因相互作用力过大而飞走。如不考虑重力作用，在第一步建立堆积模型后，采用 `reduceGravity` 命令消减重力作用，可减少上述的应力突变。

(2) 在模型加载过程中，当压力太大或颗粒刚度太小时，颗粒变形过大，导致颗粒被完全“压扁”，穿过并飞出边界。

(3) 在二维模型中，前后边界会被取消，所有单元的 Y 坐标都必须为 0。如出现大量单元飞出四个边界，需要检查单元的 Y 坐标是否为 0，特别是导入结构体后，要确认新加单元的 Y 坐标都为 0。建议可以在迭代计算前，加上 `d.mo.aY(:)=0` 命令，命名所有单元 Y 坐标为 0。

6. Q: `d.Mats` 中记录的材料微观力学参数与 `d.mo` 中的记录的微观力学参数为什么不同？

A: 材料微观力学参数与单元半径相关（见第一章转换公式和 3.4.1 节），`d.Mats` 中记录的是建立指定单元直径时（如 `Mats{1}.d`）的单元参数，`d.mo` 记录了每个颗粒的微观参数且用于实际计算。

7. Q: MatDEM 的窗口界面可以修改吗？

A: MatDEM 中所有的窗口元件都可以在程序中修改。先输入 `clc`，数据栏会显示 `handles`，其中包括所有的窗口控件，然后用 Matlab 的 `set` 命令可以修改窗口的各种参数，例如 `set(handles.figure1,'Name','My MatDEM')` 命令可以修订窗口左上的名字。通过这种方法，可以在 MatDEM 中自行增加按钮或新的窗口，具体帮助和示例将在后面版本的 MatDEM 中提供。

8. Q: 二次开发函数循环语句中运行 return 为什么有时会出错?

A: 二次开发函数的 for, if, while 和 switch 语句中不能运行 return 语句, 会自动跳过, 可以通过 if-else 语句实现 return 的效果, 例如:

```
a=55;  
if a>100  
    return;  
end  
fs.disp('a is <=100');
```

此时在 MatDEM 中会出现错误, 可以改成如下形式:

```
a=55;  
if a<=100  
    fs.disp('a is <=100');
```

9. Q: 为什么程序定义材料泊松比不能大于 0.2。

A: MatDEM 材料的初始设置基本转换公式, 转换公式允许的最大泊松比是 0.2。对于大泊松比的岩石和材料 (如橡胶), 其泊松比通常来自于其特定的结构, 需要通过特定的堆积形成较大的泊松比 (详见 1.6.1 节)。当直接设置材料泊松比为 0.2 时, 实际得到的泊松比会大于这个值, 因此, material.rate 中泊松比对应的默认系数为 0.8 (详见 3.4 节)。

10. Q: 为什么把保存好的数据加载到软件中后, 无法在后处理中查看?

A: 在迭代计算中, 在保存文件前, 通常会使用 d.clearData 函数来压缩数据, 当加载到软件中时, 需要使用 d.calculateData 来重新计算出数据。如果原数据保存在 GPU 中, 那么还需要使用 d.mo.setGPU('off')命令。

11. Q: 为什么连续使用 d.delElement 会出错?

A: 因为 d.delElement 会将指定编号的单元删除掉, 并建立一个新 d 对象, 此时 d 中的单元编号已经变化, 需重新获得所要删除的单元编号。如果要一次删除多个组, 可以将这些组编号合成一个编号数组, 并放入 d.delElement 函数中。

12. Q: 为什么保存大文件时出如下报错:

```
>位置: f.runFileCommandCells, 行: 225  
>代码行: 33, save(['TempModel/' B.name 'l.mat'],'B','d');  
>错误: 关闭文件 D:\MatDEM1.32-2\TempModel\BoxStruct1.mat 时出错。  
文件可能已损坏。
```

A: 当单元数达到数百万时, 保存的.mat 文件可能达到 GB 级别, 由于默认的 save 命令仅支持 2GB 以下的文件。此时需要声明保存文件的版本为 v7.3, 即使用 save('abc.mat','-v7.3') 命令。同时, 在保存文件前, 也可以使用 d.clearData(1)命令来压缩数据。

13. Q: 为什么 Matlab 的 length 函数不能使用?

A: MatDEM 中可以运行 Matlab 的 length 来获得矩阵的长度。在运行代码时, Matlab 函数会被新定义参数所覆盖, 当定义 length=100 后, 则无法再通过运行 length 函数来获得矩阵长度, 并可能导致报错。

14. Q: 工程尺度的离散元建模需要大量的颗粒, 如何进行数值模拟?

对于试样尺度的离散元数值模拟, 可采用一个单元代表真实岩石的一个颗粒。而对于工程尺度的问题, 如滑坡的离散元模拟, 显然无法模拟岩石中的每个颗粒。此时, 与有限元法类似, 每个离散单元可代表一个岩块, 并需要赋以相应的接触模型和力学性质。所以, 在进行大尺度的离散元数值模拟时, 需要确定好单元的尺寸, 并设定在这个尺寸下, 单元应当具有的力学性质。详见 1.5 节。

更多问题请访问 <http://matdem.com>, 加入 MatDEM 在线讨论组。