模拟重力的二维颗粒区域填充方法

本算法拟解决用二维颗粒填充区域的问题。算法分为两个部分，第一部分是颗粒的几何摆放，第二部分是颗粒的重力沉降。下面分别介绍这两个部分。

1. 几何摆放

几何摆放的目标是使用圆形，无重叠、紧密地填充任一二维区域。本部分是对前进边法的改进算法。以下以矩形区域填充为例首先对前进边法做详细地介绍，然后再介绍对其所做的改进。

首先，生成待填充的目标颗粒集合。该集合可以是服从任意粒径分布的，大小为N的集合。然后，对集合中的颗粒逐个进行几何摆放。第一次先无放回地取出集合中的任意两个颗粒A、 B。将这两个颗粒有序、相外切地摆放到矩形区域底边的左下角，生成初始前进边。此时，前进边为由区域左边界、颗粒A、颗粒B、区域下边界、区域右边界五个成员按序组成的一个有序单链表。之后所有颗粒的摆放都是基于此前进边进行。即每次从集合中无放回地取出一个颗粒，将它以与前进边中选定的两个颗粒相外切、与前进边中其它颗粒都相外离的规则进行摆放，摆放结束后更新前进边。重复以上步骤直到使用完集合中的所有颗粒或者区域填充完成。以下对前进边法摆放的关键点，即基于前进边的新颗粒摆放方法进行详细地介绍。

由上述可知，前进边法摆放新颗粒的方法可以分解为以下三个主要步骤：计算目标颗粒位置、判断目标颗粒与前进边中所有颗粒的位置关系、前进边更新。

1. 计算目标颗粒位置

每次以前进边中相邻的两个有序颗粒，假设以A和B为基颗粒，以目标集合中选取的任一颗粒为目标颗粒，按照三个圆形两两外切的规则计算目标颗粒的位置。此步骤应当注意两点，一是基颗粒要选取前进边中相切的两个邻近颗粒，不可任意选取；二是计算得到的目标颗粒的位置可能有两个，但合适解只有其中一个。合适的判断条件为在所有颗粒的摆放过程中，A、目标颗粒、B都按顺时针或逆时针两两相外切。

1. 判断位置关系

计算得到的合适的目标颗粒与前进边中所有颗粒的位置关系有四种情况。一是目标颗粒只A和B相外切，与其它颗粒都相外离。二是目标颗粒与A之前的颗粒，假设为C相交。三是目标颗粒与B之后的颗粒，假设为D相交。四是目标颗粒既与A之前的颗粒c相交，又与B之后的颗粒D相交。

1. 更新前进边

针对上述得到的四种情况，对前进边分情况进行处理。情况一，将目标颗粒插入到前进边的A和B颗粒之间。情况二，将前进边中C和A之间的颗粒删除，不包括C，但包括A。之后以C和B为基颗粒重新执行a步骤。情况三，将前进边中B和D之间的颗粒删除，包括B，但不包括D。之后以A和D为基颗粒重新执行a步骤。情况四，针前进边中C和D之间的颗粒删除，不包括C和D。之后重新执行a步骤。

以上则是前进边法生成颗粒的基本执行过程。本算法的目标是其基础上，得到考虑颗粒间摩擦的几何摆放。即对上述步骤c中结果是情况一的颗粒，判断其是否处于几何稳定。如果稳定则断续执行，如果不很定则通过旋转将其移动到稳定状态。下面对改进部分做详细介绍。

1. 搜索邻居粒子

邻居粒子即与当前颗粒相接触的所有颗粒。接触判断条件为两颗粒圆心距与两颗粒半径和之差的绝对值小于设定的容差值。

1. 邻居粒子排序

按照邻居颗粒与当前颗粒的方位角由小到大排列。方位角为当前颗粒圆心指向邻居颗粒圆心的矢量与笛卡尔坐标系X轴正向的夹角，夹角取值范围是[0, 360]。逆时针方向为正。

1. 判断稳定

以颗粒堆积的休止角为判断条件。选取上一步方位角排序中方位角为最大和最小的两个颗粒，计算这两个邻居颗粒与当前颗粒的圆心连线与笛卡尔坐标系Y轴的夹角。将所得夹角的余弦取绝对值，大于休止角余弦值的认为当前颗粒不稳定，并记录邻居颗粒序号。

1. 若不稳定，移动到稳定态

经上一步判断得到不稳定状态的所有邻居粒子。将上一步中所得邻居粒子中夹角最大的邻居颗粒标定为旋转轴粒子。以标定的旋转轴粒子为中心，旋转当前颗粒直到与别的颗粒接触，或者与边界接触。

1. 迭代a-d，直到稳定
2. 重力沉降

重力沉降的目标是得到颗粒由于重力所致的沉降量与压缩量，进而得到模拟重力的颗粒摆放形态。本部分利用虚位移原理快速求解二维颗粒系统在重力作用下所有颗粒的位形，包含如下主要步骤。

1. 搜索邻居粒子

邻居粒子即与当前颗粒相接触的所有颗粒。接触判断条件为两颗粒圆心距与两颗粒半径和之差的绝对值小于设定的容差值。此处邻居粒子包含边界。

1. 建立系统能量方程

能量方程由弹性势能与重力势能两部分组成。将所有颗粒与其邻居颗粒两两配对成接触对，得到系统所有的无重复的接触对，包括颗粒与边界的接触。对所有接触对建立弹性势能方程。重力势能方程则是对所有颗粒的重力势能求和。这两个能量方程都以颗粒位置坐标为自变量。

1. 求解能量方程

对所得能量方程用合适的方法求解，得到所有颗粒的位置坐标。

1. 判断系统是否稳定

对上述结果重新执行a步骤，如果所有颗粒的邻居粒子与其上次得到的邻居粒子相同，则认为系统稳定，结束求解。如果不稳定，则断续执行直到稳定。