

ZDEM 参考手册

v2.x

李长圣

2024 年 4 月 5 日

目 录

第 1 章 简介	4
第 2 章 如何使用	6
第 3 章 一个示例学会 ZDEM	7
第 4 章 提交计算	10
4.1 直接提交	10
4.2 Slurm 作业调度系统	10
第 5 章 构造模拟	13
5.1 同构造剥蚀	13
5.2 同构造沉积	16
5.3 先存断层设置	18
5.4 滑脱层	20
5.5 古隆起	22
5.6 刚性基底伸展构造	24
5.7 与物理模拟对比实验	28
5.8 韧性基底伸展构造	31
第 6 章 应力应变	37
6.1 配置环境变量	37
6.2 无沉积剥蚀	38
6.3 有沉积剥蚀	40
6.3.1 流程解析	40
6.3.2 实例	44
6.4 绘制最大主应力矢量	48
第 7 章 命令参考	51
第 8 章 离散元原理	69
8.1 最小的离散元程序	69
8.2 颗粒位置的更新	71
8.3 接触力的计算	72
8.3.1 线弹性模型	73
8.3.2 Hertz-Mindlin 模型	73

第 9 章 颜色表	74
第 10 章 开通账户	75
第 11 章 Linux 命令行	76
第 12 章 使用协议	78
第 13 章 版本更新	80
第 14 章 致谢	81

欢迎来到 [离散元](#) 的世界。

首先来看一个示例学会 [ZDEM](#)

本项目是由 [李长圣科研团队](#) 维护的 ZDEM 参考手册, 即可以作为 ZDEM 日常参考, 也可以作为离散元的入门读物。希望通过阅读本手册, 能够让用户尽快掌握软件使用方法。

脚本约定

- 不区分大小写, 不支持任何中文字符。
- () = , 和空格都会被忽略
- # ! ; 均是注释符, 注释符后面的所有内容 (可以使用中文) 会被忽略。
- 程序必须以 START、RESTORE 或 LOAD 开始一个计算。

初级视频教程

- [geovbox@ 哔哩哔哩](#)

相关链接:

- 官网: <https://geovbox.com>
- 手册: <https://doc.geovbox.com>

第 1 章 简介

- ZDEM 是什么

ZDEM，是一个用于构造变形模拟的二维离散元数值模拟软件。采用 C 语言编写，并用 OpenMP 完成了并行设计。主要面向构造模拟，用来补充构造物理沙箱实验在应力应变及材料选取上的局限性，为构造变形研究提供一种新的方法。

- ZDEM 的历史

- 2021 年 4 月 19 日, ZDEM 2.0 发布;
- 2021 年 6 月 17 日, ZDEM 2.1 发布;
- 2022 年 5 月 11 日, ZDEM 2.2 发布;
- 目前最新版本 ZDEM 2.2 发布于 2022-05-11。

- 开发者

源码由 [李长圣](#) 开发并负责维护。



图 1.1: 软件维护者

- 软件的特点

为什么选择 ZDEM 作为构造模拟呢?

1. 使用方便

登陆 [并行超算云](#)，既可使用。

2. 使用门槛低

命令参数采用 PFC4.0 类似参数，并配有丰富的测试、休止角、三轴、构造模拟等相关实例。

3. 面向构造变形研究

针对构造中的速度不连续面等编写了相应的模块，让构造模拟更方便。

4. 高性能

采用 C 语言, 基于 OpenMP 完成并行设计。

5. 跨平台

支持浏览器登陆, windows、linux 和 mac 操作系统都可以使用。

6. 模块化

遵循模块化设计思想, 将不同的接触力学模型划分到不同的模块中。当前支持固体晶格模型, 线弹模型及赫兹模型。这样的模块化设计有很多优点:

- 只需要少量的模块
- 各个模块之间相互独立且代码量少, 易于更新和维护

7. 支持多种格式的高精度矢量图和位图

支持多种高精度的矢量图片格式和位图图片格式。

- VTK 格式: [ParaView](#) 支持, 可以直接观看制作动画, 修改配色等其他操作
- 矢量图片格式, 如 PDF、PS、EPS 和 SVG, 具有任意放大缩小而不失真的特性, 可直接投稿到学术期刊
- 位图图片格式, 如 BMP、JPG、PNG、PPM 和 TIFF 格式, 可用于日常的文档及演示

• 同类产品

在构造模拟方面, 还有一些软件也可以实现类似的功能。

1. VBOX: [VBOX1.4](#) 暂停更新
2. PFC2D: [PFC2D](#)
3. Yade: [手册](#) | [代码下载](#)
4. MatDEM: [MatDEM](#)
5. DICE2D: [DICE2D](#)

待处理: 介绍更多同类产品, 如 TRUBAL, RICEBAL 等。

第 2 章 如何使用

备注： 使用浏览器 (支持 windows、linux 和 mac 操作系统) 登陆 [并行超算云](#) 即可提交计算

1. 申请并行超算云帐号, 办法见[开通账户](#)
2. 打开浏览器输入登陆并行超算云 <https://cloud.paratera.com/>
3. 输入申请的用户名和密码, 即可登陆。

第 3 章 一个示例学会 ZDEM

一个构造沙箱试验仿真的完整流程：

- 生成。生成颗粒集合体，定义颗粒的材料参数，让其在重力作用下沉积，形成初始模型。
- 挤压。给定墙相应的速度，开始挤压。

下面是一个最简单的挤压计算实例，26 行命令即可完成该计算。学会了该命令脚本，基本掌握 ZDEM 的使用方法。

登陆 [并行超算云](#)，运行 `push.py` 脚本：

```
zdem push.py
```

`push.py` 中完整脚本命令如下

```
#####
# title: 一个实例学会 ZDEM
# date: 2021-04-25
# authors: 李长圣
# E-mail: sheng0619@163.com
# note:
# 括号内参数可根据模型大小及个人需要修改
# 脚本命令不区分大小写
# 用 16 个核心，实际用时 < 1 小时
# 计算费用约 < 2 元
# more info, see www.geovbox.com
#####
#程序初始化
START
#颗粒设为球，计算颗粒体积用  $4/3\pi r^3$  计算
SET disk off
#设置研究范围
BOX left 0.0 right 41000.0 bottom 0.0 height 11000.0 kn=0e10 ks=0e10 fric 0.00
#设置挡板墙，这里模型采用 hertz 接触模型，挡板墙的 kn ks 无效，计算时取颗粒的参数
WALL ID 0, NODES ( 0.0 , 10.0 ) ( 40000.0 , 10.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR_
↪black
WALL ID 1, NODES ( 10.0 , 10000.0 ) ( 10.0 , 10.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR_
↪blue
WALL ID 2, NODES ( 40000.0 , 10.0 ) ( 40000.0 , 10000.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR red
#在矩形范围内生成颗粒
GEN NUM 100000 rad discrete 60.0 80.0, x ( 10.0, 40000.0), y ( 10.0, 10000.0), COLOR black GROUP_
↪ball_rand
#设置颗粒的微观参数 density 密度, fric 摩擦系数, shear 剪切模量, poiss 泊松比, damp_
↪局部阻尼常数, heart Hertz-Mindlin 接触模型
```

(续下页)

(接上页)

```

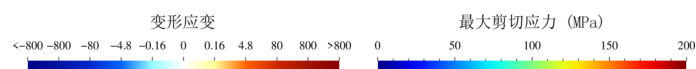
PROP density 2.5e3, fric 0.0, shear 2.9e9, poiss 0.2, damp 0.4, hertz
#设置时间步及重力加速度
SET DT 5e-2, GRAVITY 0.0, -9.8
#设置每1000步保存一次vtk格式的计算结果
SET vtk 1000
#设置每1000步保存一次ps格式的计算结果
SET ps 1000
#设置每1000步保存一次dat格式的计算结果
SET print 1000
#沉积, 计算5000步
CYC 5000
#删除4000米以上的颗粒
DEL RANGE y 4000.0 999000.0
#平衡, 计算1000步
CYC 1000
#输出包含颗粒的[x y r]信息的初始模型 init_xyr.dat
#EXP init_xyr.dat

#设置bond粘结, 使颗粒具有粘聚力, ebmod 杨氏模量, gbmod 剪切模量, tstrength
↔抗拉强度, sstrength 聚合强度, fric 摩擦系数
PROP ebmod 2e8 gbmod 2e8 tstrength 2e7 sstrength 4e7 fric 0.3
#给地层赋上颜色
PROP COLOR lg range y 0.0 500.0
PROP COLOR green range y 500.0 1000.0
PROP COLOR yellow range y 1000.0 1500.0
PROP COLOR red range y 1500.0 2000.0
PROP COLOR black range y 2000.0 2500.0
PROP COLOR mg range y 2500.0 3000.0
PROP COLOR blue range y 3000.0 3500.0
PROP COLOR gb range y 3500.0 4000.0
PROP COLOR violet range y 4000.0 4500.0

#设置挡板墙摩擦系数
WALL id 0 fric 0.3
WALL id 1 fric 0.3
WALL id 2 fric 0.3
#设置墙的挤压速度 x方向速度为2.0
WALL id 1 xv 2.0
#设置墙的挤压量x方向推进10000.0, 每挤压2000.0保存一次计算结果
IMPLE wall id 1 xmove 10000.0 save 2000.0 print 1000.0 ps 1000.0 vtk 1000.0
#计算停止
STOP

```

计算结束后, 将得到以下结果:



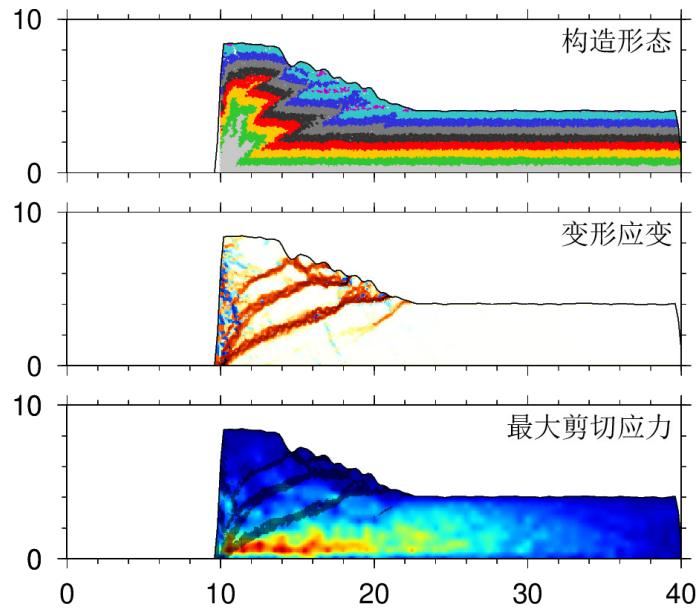


图 3.1: 单位 (km)

警告: 软件中, 坐标不支持负数, 为了防止墙体和 BOX 重合, 本示例将模型左下角起始点为 $(10.0, 10.0)$ 。实际设计实验时, 一般将模型左下角起始点设置为 $(1000.0, 1000.0)$, 再结合 `zdemss` 中的 `--xmove=-1000.0` 和 `--ymove=-1000.0` 将模型左下角移到 $(0.0, 0.0)$ 。

颗粒参数表

颗粒直径 $d(m)$	颗粒密度 $(kg \cdot m^{-3})$	颗粒摩擦系数 μ	缩短速率 $v(m \cdot s^{-1})$	时间步 s
60, 80	2500	0.3	2.0	0.05

第 4 章 提交计算

push.py 为要运行的脚本名

4.1 直接提交

```
#程序名 脚本名  
zdem push.py
```

- 优点: 实时查看计算是否正确, 使用 `ctrl+c` 结束计算
- 缺点: 关闭窗口, 计算停止

警告: 这种方式只能作简单测试用, 长时间使用会被管理员封号。如果想交互式的提交计算, 可采用如下命令 `srun -n 1 -c 12 zdem push.py`, 即采用 1 个节点 `-n 1`, 12 个核 `-c 12` 计算。这种方式会正常计费, 不会被管理员封号。

4.2 Slurm 作业调度系统

[并行超算云](#) 安装了该调度系统。

SLURM (Simple Linux Utility for Resource Management) 是一种可扩展的工作负载管理器, 已被全世界的国家超级计算机中心广泛采用。它是免费且开源的, 根据 GPL 通用公共许可证发行。

- `sbatch job.sh` 提交作业
- `squeue` 查看作业状态 (可以查看所有作业的 id 号)
- `scancel id` 取消一个作业 (id 号定位该作业)

备注:

- 优点: 同时计算多个, 关闭窗口, 不影响计算
 - 缺点: 不能实时查看计算情况
-

4.2.1 提交作业 sbatch job.sh

显示

```
Submitted batch job 134573
```

说明提交成功。`job.sh` 内容如下:

```
#!/bin/bash
#SBATCH --job-name=test
#SBATCH --partition=v6_384
#SBATCH -n 1
#SBATCH -c 12
#SBATCH -t 1440
#SBATCH --output=%j.out
#SBATCH --error=%j.err

source /public1/soft/modules/module.sh
source /public1/soft/other/module_zdem.sh
module load zdem2.0

time srun -n 1 zdem push.py
time srun -n 1 zdem2jpg --dir=./data
```

警告: 这里配置 `module load zdem2.0` , 将使用 2.* 系列的最新版本。

警告: 注意 `job.sh` 的行结尾符需采用 Unix 的 `\n` 格式, 否则将产生如下错误信息:

```
sbatch: error: Batch script contains DOS line breaks (\r\n)
sbatch: error: instead of expected UNIX line breaks (\n).
```

修改 `job.sh` 的行结尾符见 [第 00 课并行超算云上使用 ZDEM 的 10:15](#)

4.2.2 查看作业状态 squeue

显示

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST (REASON)
134573	v6_384	test	sc80502	R	0:30	1	cb0503

从上面的输出可知, 该任务的 id 为 134573。ST 状态为 R 说明正在计算。任务分配给了节点 cb0503。

小技巧: `squeue -l` 可以查看作业细节信息

4.2.3 取消作业 `scancel id`

要杀死上面的作业, 输入 `scancel 134573` 即可, 134573 为上面作业的 id 号。

再次输入 `squeue`。显示

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST	(REASON)
-------	-----------	------	------	----	------	-------	----------	----------

说明作业已经取消。

参见:

Slurm 更多介绍 <https://docs.hpc.sjtu.edu.cn/job/slurm.html>

第 5 章 构造模拟

本章我们将给出构造模拟中经常涉及到的一些示例, 如同构造剥蚀 同构造剥蚀

5.1 同构造剥蚀

这里是一个 同构造剥蚀计算实例。

`syn_erosion.py` 和 `0000046000.sav` 需放在同一目录下。

`syn_erosion.py` 中完整脚本命令如下

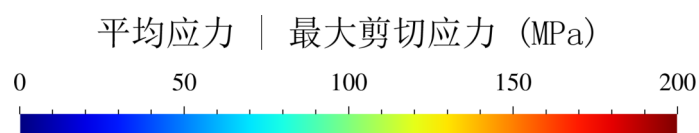
```
#####
# title: 同构造沉积
# date: 2019-01-19
# authors: 李长圣
# E-mail: sheng0619@163.com
# more info, see www.geovbox.com
#####
#从 0000046000.sav <版本1.3> 计算节点恢复, 由 一个实例学会 ZDEM 生成
RES 0000046000.sav
# 每次 100 步更新一次进度条
SET stepbar 100
#设置墙的挤压速度 x方向速度为2.0
WALL id 1 xv 2.0
#设置墙的挤压量x方向推进1000.0, 每挤压1000.0保存一次计算结果
IMPLE wall id 1 xmove 1000.0 save 1000.0 print 1000.0 ps 1000.0

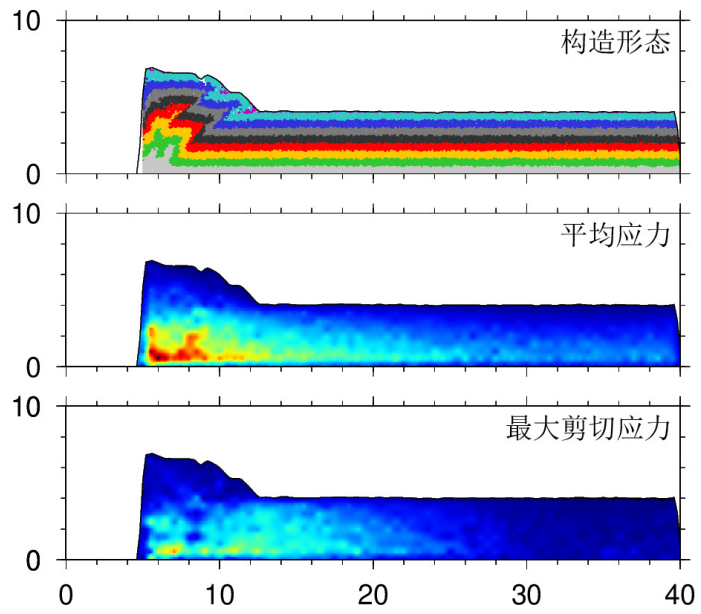
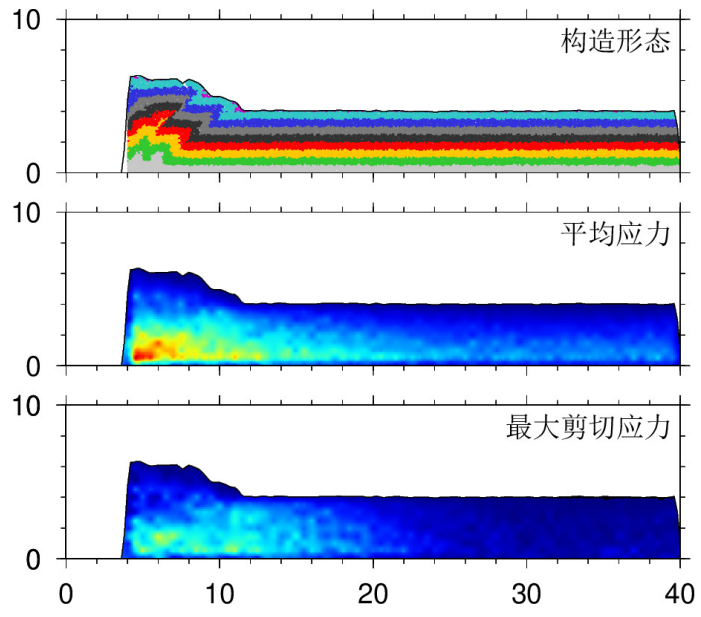
##### 剥蚀 #####
#删除4000米以上的颗粒
DEL RANGE y 4000.0 999000.0
#####

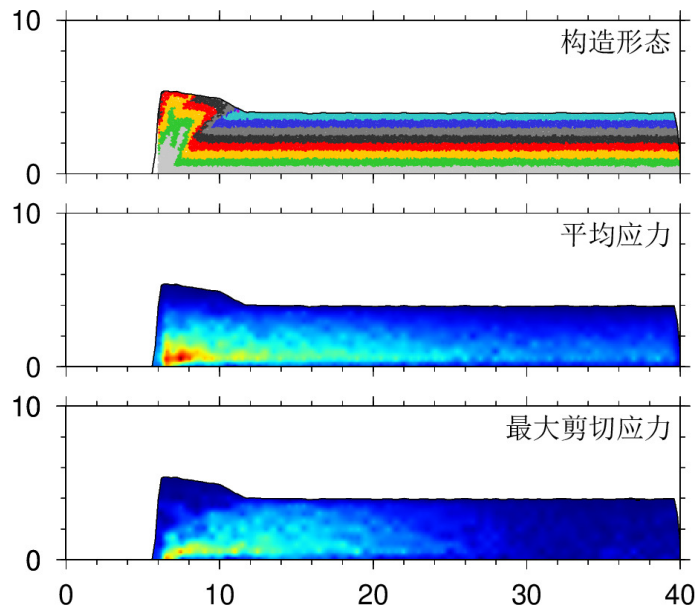
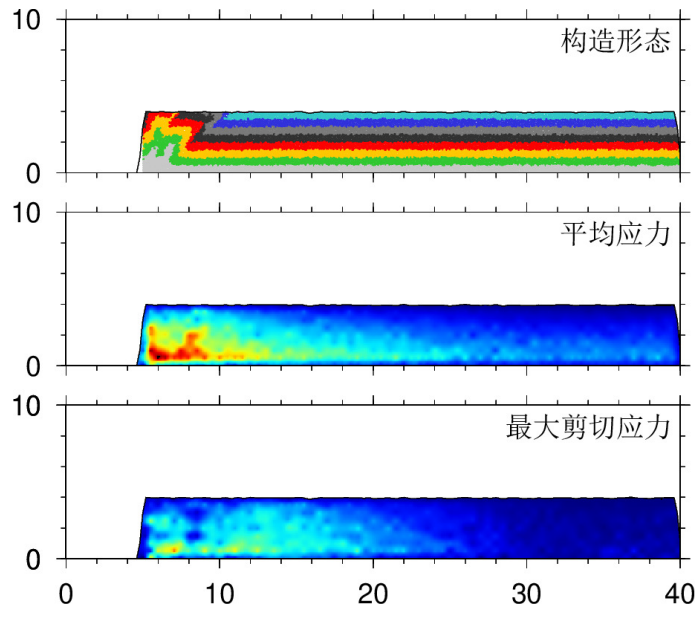
#设置墙的挤压速度 x方向速度为2.0
WALL id 1 xv 2.0
#设置墙的挤压量x方向推进5000.0, 每挤压2000.0保存一次计算结果
IMPLE wall id 1 xmove 5000.0 save 2000.0 print 1000.0 ps 1000.0

#计算停止
STOP
```

计算结束后, 将得到以下结果:







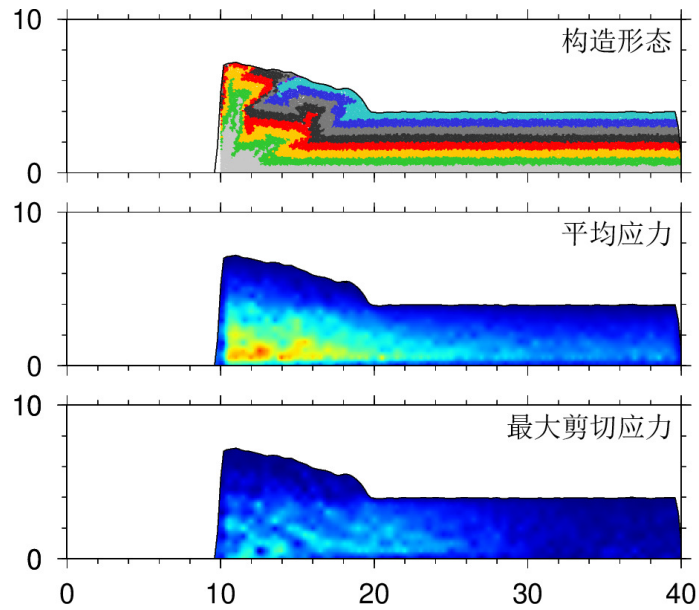


图 5.1: 单位 (km)

5.2 同构造沉积

这里是一个同构造沉积计算实例。

`syn_sedimentaion.py` 和 `0000046000.sav` 需放在同一目录下。

`syn_sedimentaion.py` 中完整脚本命令如下

```
#####
# title: 同构造沉积
# date: 2019-01-19
# authors: 李长圣
# E-mail: sheng0619@163.com
# more info, see www.geovbo.com
#####
#从 0000046000.sav <版本1.3> 计算节点恢复, 由 一个实例学会 ZDEM 生成
RES 0000046000.sav
# 每次 100 步更新一次进度条
SET stepbar 100
#设置墙的挤压速度  $x$ 方向速度为2.0
WALL id 1 xv 2.0
#设置墙的挤压量 $x$ 方向推进1000.0, 每挤压2000.0保存一次计算结果
IMPLE wall id 1 xmove 1000.0 save 2000.0 print 1000.0 ps 1000.0

##### 沉积 #####
#停止挤压, 墙的 $x$ 方向速度改为0.0
WALL id 1 xv 0.0
#沉积。在挤压前端12000~40000.0上方, 沉积约 1 km 颗粒。 $y$ 的范围需要设置为4000-6000。
#经验: 颗粒充填2km范围, 沉积之后的地层厚度约为1km
GEN NUM 100000.0 rad discrete 60.0 80.0, x ( 12000.0, 40000.0), y ( 4000.0, 6000.0), COLOR red
↔GROUP sed
#设置沉积颗粒 GROUP=sed 的微观参数
```

(续下页)

(接上页)

```

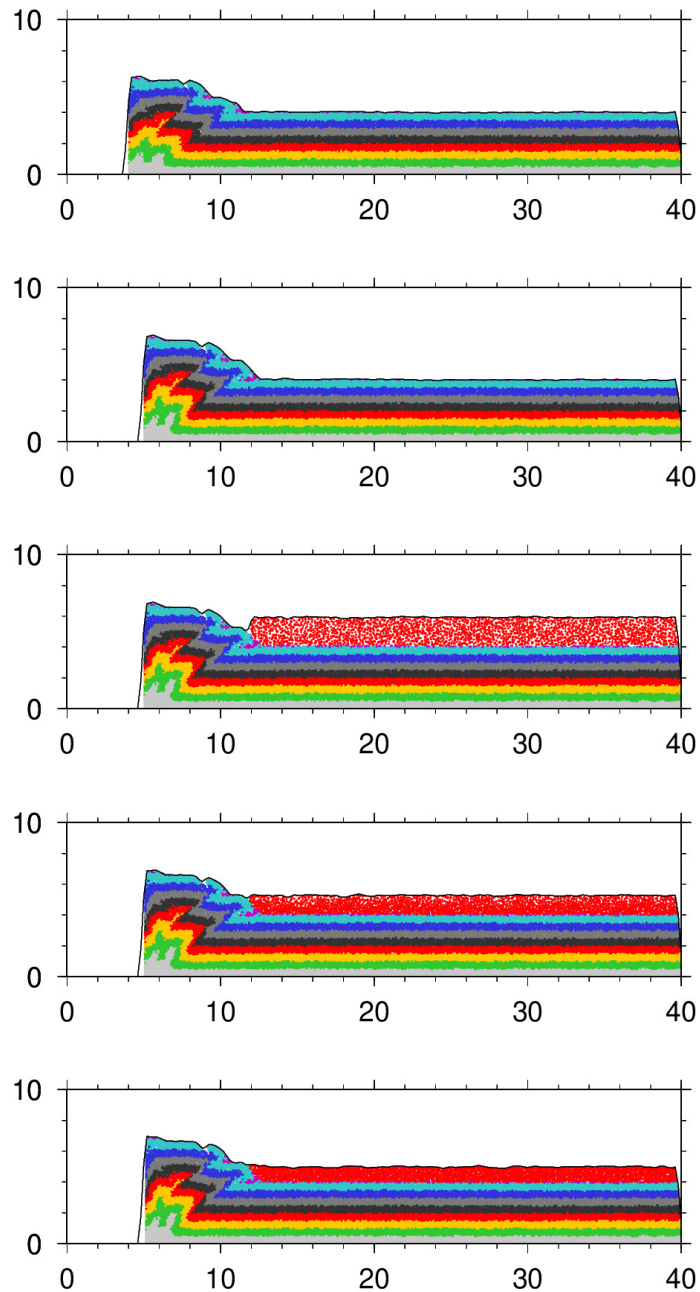
PROP DENSITY 2.5e3, fric 0.3, shear 2.9e9, poiss 0.2, damp 0.4, hertz range GROUP sed
#计算2000步, 让颗粒沉积下来
SET print 100 #每 100 步输出一一次计算结果
CYC 2000
#####

#设置墙的挤压速度 x方向速度为2.0
WALL id 1 xv 2.0
#设置墙的挤压量x方向推进5000.0, 每挤压2000.0保存一次计算结果
IMPLE wall id 1 xmove 5000.0 save 2000.0 print 1000.0 ps 1000.0

#计算停止
STOP

```

计算结束后, 将得到以下结果:



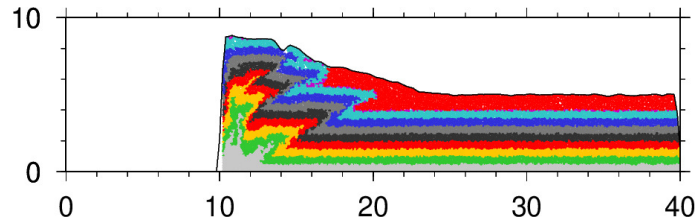


图 5.2: 单位 (km)

5.3 先存断层设置

这里是一个 断层设置实例。

pre_struct.py 中完整脚本命令如下

```
#####
# title: 断层设置方法
# date: 2019-01-20
# authors: 李长圣
# E-mail: sheng0619@163.com
# more info, see www.geovbox.com
#####
#程序初始化
START
#关闭圆盘，颗粒设为球，计算颗粒体积用 $4/3\pi r^3$ 计算
set disk off
#设置研究范围
BOX left 0.0 right 41000.0 bottom 0.0 height 11000.0 kn=0e10 ks=0e10 fric 0.00
#设置挡板墙，这里模型采用hertz接触模型，挡板墙的kn ks无效，计算时取颗粒的参数
WALL ID 0, NODES ( 0.0 , 10.0 ) ( 40000.0 , 10.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR_
↪black
WALL ID 1, NODES ( 10.0 , 10000.0 ) ( 10.0 , 10.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR_
↪blue
WALL ID 2, NODES ( 40000.0 , 10.0 ) ( 40000.0 , 10000.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR red
#在矩形范围内生成颗粒
GEN NUM 100000.0 rad discrete 60.0 80.0, x ( 10.0, 40000.0), y ( 10.0, 10000.0), COLOR black GROUP_
↪ball_rand
#设置颗粒的微观参数
PROP DENSITY 2.5e3, fric 0.0, shear 2.9e9, poiss 0.2, damp 0.4, hertz
#设置时间步及重力加速度
SET DT 5e-2, GRAVITY 0.0, -9.8
#设置每1000步保存一次dat格式的计算结果
SET print 1000
#沉积，计算5000步
CYC 5000
#删除4000米以上的颗粒
DEL RANGE y 4000.0 999000.0
#平衡，计算1000步
CYC 1000
#输出包含颗粒的[x y r]信息的初始模型 init_xyr.dat
EXP init_xyr.dat

#设置bond粘结，使颗粒具有粘聚力
```

(续下页)

(接上页)

```

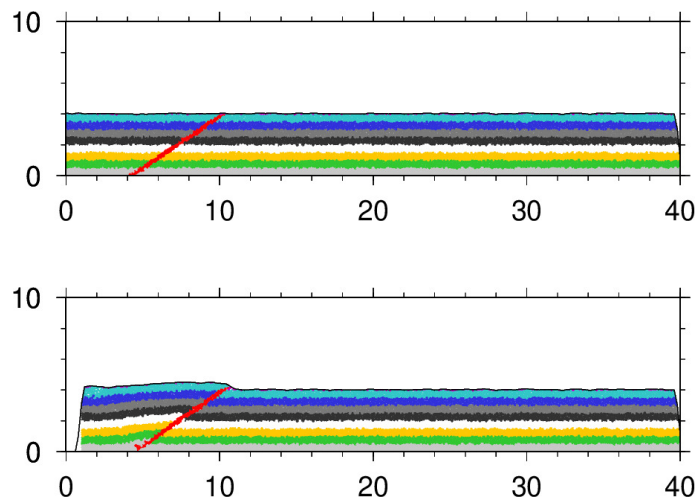
PROP ebmod 2e8 gbmod 2e8 tstrength 2e7 sstrength 4e7 fric 0.3
#给地层赋上颜色
PROP COLOR lg      range y  0.0  500.0
PROP COLOR green   range y 500.0 1000.0
PROP COLOR yellow  range y 1000.0 1500.0
PROP COLOR white   range y 1500.0 2000.0
PROP COLOR black   range y 2000.0 2500.0
PROP COLOR mg      range y 2500.0 3000.0
PROP COLOR blue    range y 3000.0 3500.0
PROP COLOR gb      range y 3500.0 4000.0
PROP COLOR violet  range y 4000.0 4500.0

##### 断层设置 #####
#用range P4 (point1) (point2) (point3) (point4) 命令, 逆时针指定四个点
#四个点组成的多边形, 设置为组 struct1
PROP GROUP struct1 RANGE P4 (4000.0, 0.0) (4500.0, 0.0) (10500.0 4000.0) (10000.0 4000.0)
#打断struct1组内的颗粒粘结
BOND break RANGE GROUP struct1
#将struct1组的颗粒颜色设置为红色, 摩擦系数设置为0.0, 摩擦系数可以根据断层强弱改变
PROP COLOR red FRIC 0.0 RANGE GROUP struct1
#####

#设置挡板墙摩擦系数
WALL id 0 fric 0.3
WALL id 1 fric 0.3
WALL id 2 fric 0.3
#设置墙的挤压速度 x方向速度为2.0
WALL id 1 xv 2.0
#设置墙的挤压量x方向推进4000.0, 每挤压1000.0保存一次计算结果
IMPLE wall id 1 xmove 4000.0 save 1000.0 print 1000.0 ps 1000.0
#计算停止
STOP

```

计算结束后, 将得到以下结果:



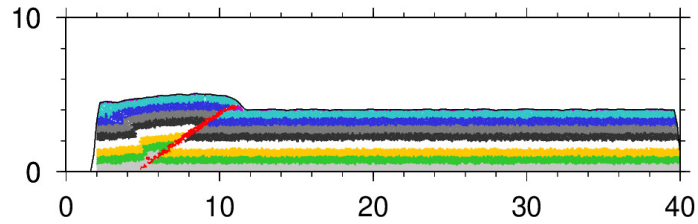


图 5.3: 单位 (km)

5.4 滑脱层

这里是一个滑脱层设置的实例，一般地，将滑脱层设置为没有粘结，类似松散石英砂，摩擦系数则可以设置为 0.0。

`detachment.py` 和 `init_xyr.dat` 需放在同一目录下。

`detachment.py` 中内容如下：

```
#####
# title: 滑脱层设置方法
# date: 2019-06-29
# authors: 李长圣
# E-mail: sheng0619@163.com
# more info, see www.geovboz.com
#####
# init_xyr.dat 保存了沉积后的所有颗粒的 [x y r]信息，由一个实例学会VBOX生成
LOAD init_xyr.dat
#导入的颗粒命名为 ball_rand
PROP GROUP ball_rand
#关闭圆盘，颗粒设为球，计算颗粒体积用 $4/3\pi r^3$ 计算
SET DISK OFF
#设置研究范围
BOX LEFT 0.0 RIGHT 41000.0 BOTTOM 0.0 HEIGHT 11000.0 KN=0e10 KS=0e10 FRIC=0.00
#设置挡板墙，这里模型采用hertz接触模型，挡板墙的kn ks无效，计算时取颗粒的参数
WALL ID 0, NODES ( 0.0 , 10.0 ) ( 40000.0 , 10.0 ), KN=0e10 KS=0e10 FRIC=0.0 COLOR_
↪black
WALL ID 1, NODES ( 10.0 , 10000.0 ) ( 10.0 , 10.0 ), KN=0e10 KS=0e10 FRIC=0.0 COLOR_
↪blue
WALL ID 2, NODES ( 40000.0 , 10.0 ) ( 40000.0 , 10000.0 ), KN=0e10 KS=0e10 FRIC=0.0 COLOR red

#设置颗粒的微观参数
PROP DENSITY 2.5e3, FRIC 0.0, SHEAR 2.9e9, POISS 0.2, DAMP 0.4, HERTZ
#设置时间步及重力加速度
SET DT 5e-2, GRAVITY 0.0, -9.8

#设置bond粘结，使颗粒具有粘聚力
PROP ebmod 2e8 gbmod 2e8 tstrength 0e7 sstrength 0e7 fric 0.3
#给地层赋上颜色
PROP COLOR lg range y 0.0 500.0
PROP COLOR green range y 500.0 1000.0
PROP COLOR yellow range y 1000.0 1500.0
PROP COLOR white range y 1500.0 2000.0
PROP COLOR black range y 2000.0 2500.0
```

(续下页)

(接上页)

```

PROP COLOR mg          range y 2500.0 3000.0
PROP COLOR blue        range y 3000.0 3500.0
PROP COLOR gb          range y 3500.0 4000.0
PROP COLOR violet      range y 4000.0 4500.0

##### 滑脱层设置 #####
#用 range y (ymin ymax) 圈定范围, 该范围内颗粒命名为 detachment
PROP GROUP detachment RANGE y ( 0.0, 1000.0)
#打断 detachment 组内的颗粒粘结
BOND break RANGE GROUP detachment
#将 detachment 组的颗粒颜色设置为红色, 摩擦系数设置为 0.0, 摩擦系数可以根据滑脱层强弱改变
PROP COLOR red FRIC 0.0 DEN 2.2e3 RANGE GROUP detachment
#####

#设置挡板墙摩擦系数
WALL ID 0 fric 0.0
WALL ID 1 fric 0.3
WALL ID 2 fric 0.3
#设置墙的挤压速度 x 方向速度为 2.0
WALL ID 1 XV 2.0
#设置墙的挤压量 x 方向推进 10000.0, 每挤压 1000.0 保存一次计算结果
IMPLE WALL ID 1 XMOVE 10000.0 SAVE 5000.0 PRINT 1000.0 PS 5000.0
#计算停止
STOP

```

计算结束后, 将得到以下结果:

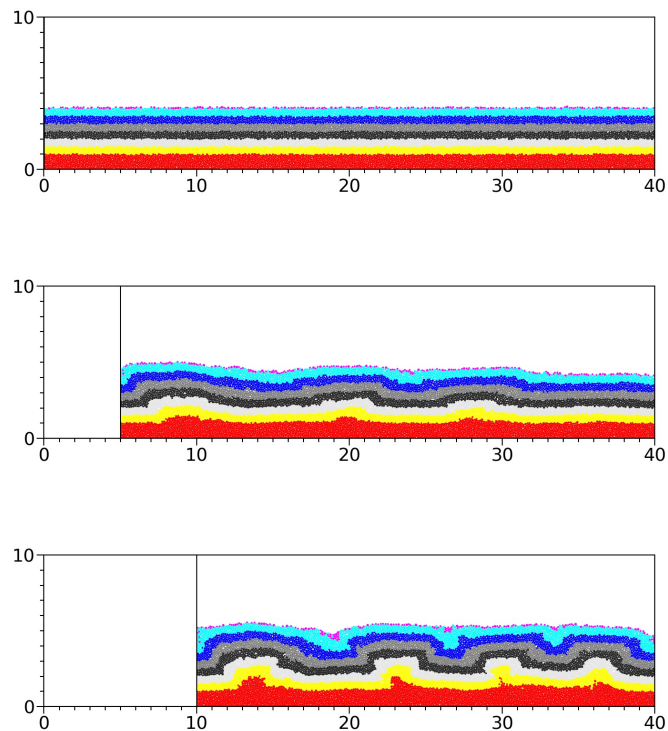


图 5.4: 单位 (km)

参考文献

1. 李长圣, 尹宏伟. 滑脱层强度对挤压构造的影响: 离散元数值模拟 [A]. 2017 中国地球科学联合学术年会论文集, 2017:4.

5.5 古隆起

这里是一个 古隆起设置的实例, 将古隆起的 x, y, spin (角速度) 固定。

`palaeohigh.py` 和 `init_xyr.dat` 需放在同一目录下。

`palaeohigh.py` 中内容如下:

```
#####
# title: 古隆起设置方法
# date: 2020-08-11 添加 INI, 初始化古隆起颗粒速度为零
# date: 2019-06-29
# authors: 李长圣
# E-mail: sheng0619@163.com
# more info, see www.geovbox.com
#####
#程序初始化
LOAD init_xyr.dat
#导入的颗粒命名为 ball_rand
PROP GROUP ball_rand
#关闭圆盘, 颗粒设为球, 计算颗粒体积用 $4/3\pi r^3$ 计算
SET disk off
#设置研究范围
BOX left 0.0 right 41000.0 bottom 0.0 height 11000.0 kn=0e10 ks=0e10 fric 0.00
#设置挡板墙, 这里模型采用 hertz 接触模型, 挡板墙的 kn ks 无效, 计算时取颗粒的参数
WALL ID 0, NODES ( 0.0 , 10.0 ) ( 40000.0 , 10.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR_
↪black
WALL ID 1, NODES ( 10.0 , 10000.0 ) ( 10.0 , 10.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR_
↪blue
WALL ID 2, NODES ( 40000.0 , 10.0 ) ( 40000.0 , 10000.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR red
#设置颗粒的微观参数
PROP DENSITY 2.5e3, FRIC 0.0, SHEAR 2.9e9, POISS 0.2, DAMP 0.4, HERTZ
#设置时间步及重力加速度
SET DT 5e-2, GRAVITY (0.0, -9.8)

#设置 bond 粘结, 使颗粒具有粘聚力
PROP ebmod 2e8 gbmod 2e8 tstrength 0e7 sstrength 0e7 fric 0.3
#给地层赋上颜色
PROP COLOR lg range y 0.0 500.0
PROP COLOR green range y 500.0 1000.0
PROP COLOR yellow range y 1000.0 1500.0
PROP COLOR white range y 1500.0 2000.0
PROP COLOR black range y 2000.0 2500.0
PROP COLOR mg range y 2500.0 3000.0
PROP COLOR blue range y 3000.0 3500.0
PROP COLOR gb range y 3500.0 4000.0
PROP COLOR violet range y 4000.0 4500.0

##### 隆起设置 #####
#用 ellipse 指定椭圆, 椭圆的中心 (20000.0, 0.0) 长轴 2000.0 短轴 1000.0
PROP GROUP palaeohigh RANGE ellipse ( 20000.0, 0.0) 2000.0, 1000.0
```

(续下页)

(接上页)

```

#用range P4 (point1) (point2) (point3) (point4) 命令, 逆时针指定四个点
#四个点组成的多边形, 设置为组 palaeohigh
PROP GROUP palaeohigh RANGE P4 ( 10000.0, 0.0) ( 12000.0, 0.0) ( 15000.0 0.0) ( 12500.0 1500.0)
#打断palaeohigh组内的颗粒粘结
BOND break RANGE GROUP palaeohigh
#初始化古隆起颗粒速度为零
INI xv 0.0 yv 0.0 spin 0.0 RANGE GROUP palaeohigh
#限制古隆起颗粒运动
FIX x y spin RANGE GROUP palaeohigh
#将palaeohigh组的颗粒颜色设置为黑色
PROP COLOR black RANGE GROUP palaeohigh
#####

#设置挡板墙摩擦系数
WALL id 0 fric 0.0
WALL id 1 fric 0.3
WALL id 2 fric 0.3
#设置墙的挤压速度 x方向速度为2.0
WALL id 1 xv 2.0
#设置墙的挤压量x方向推进10000.0, 每挤压1000.0保存一次计算结果
IMPLE WALL id 1 xmove 10000.0 save 1000.0 print 1000.0 ps 1000.0
#计算停止
STOP

```

计算结束后, 将得到以下结果:

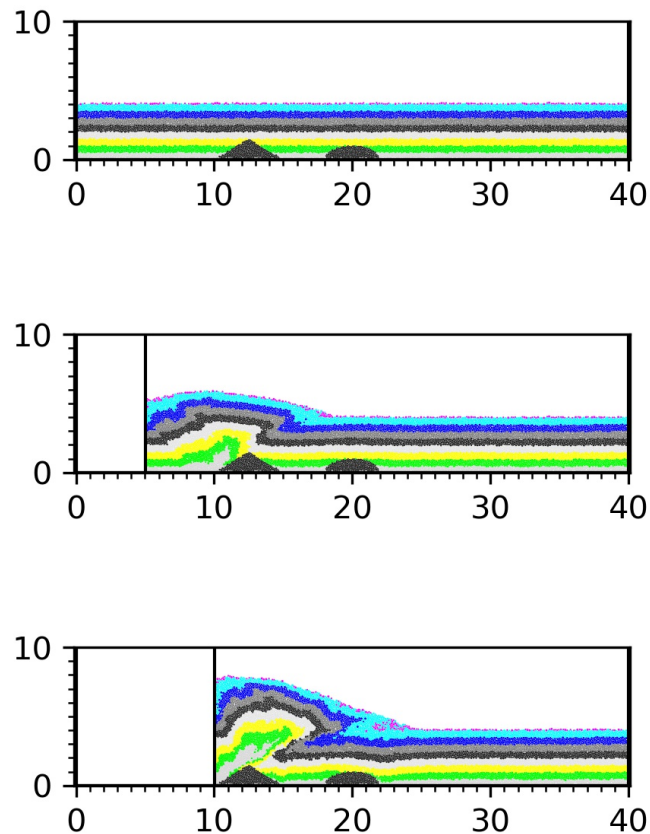


图 5.5: 单位 (km)

5.6 刚性基底伸展构造

这里是一个 刚性基底伸展构造设置的实例，使用上下两个刚性板完成。

`extens_rigid.py` 中内容如下：

```
#2019-06-29
#建议使用notepad++查看本文件
#LI ChangSheng @ NanJing Uniniversity
#E-mail: sheng0619@163.com
#more info, see www.geovbox.com
#####
# title: 刚性基底伸展构造实例
#括号内参数可根据模型大小及个人需要修改
#脚本命令不区分大小写
#该计算在南京大学高性能计算中心
#####
#程序初始化
START
#颗粒设为球，计算颗粒体积用 $4/3\pi r^3$ 计算
SET DISK off
#设置研究范围
BOX LEFT 1.0e-3 RIGHT (51000.0) BOTTOM 1.0e-3 HEIGHT (26000.0) kn=4e10 ks=4e10 fric 0.30
#设置挡板墙，这里模型采用hertz接触模型，挡板墙的kn ks无效，计算时取颗粒的参数
WALL ID 0, NODES ( 10000.0 , 26000.0 ) ( 10000.0 , 1080.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR_
↔black
WALL ID 1, NODES ( 40000.0 , 1080.0 ) ( 40000.0 , 26000.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR_
↔blue
#设置基底
GLINE P1 ( 10000.0 , 1160.0 ) P2 ( 20000.0 , 1160.0 ) RAD 80.0 GROUP bom_wall1
GLINE P1 ( 10000.0 , 1000.0 ) P2 ( 40000.0 , 1000.0 ) RAD 80.0 GROUP bom_wall2
PROP COLOR blue RANGE GROUP bom_wall1
PROP COLOR red RANGE GROUP bom_wall2
#限制颗粒位移和转动
FIX x y spin RANGE GROUP bom_wall1
FIX x y spin RANGE GROUP bom_wall2
#在矩形范围内生成颗粒
GEN NUM 100000, RAD DISCRETE (60.0 80.0) x (10000.0, 40000.0) y (1000.0, 26000.0) GROUP ball_rand
PROP COLOR black RANGE GROUP ball_rand
PROP DEN 2.5e3 FRIC 0.0 SHEAR 2.9e9 POISS 0.2 DAMP 0.4 HERTZ

SET STEPBAR 1000
SET SAVE 20000
set PRINT 10000
set PS 10000

SET DT 5e-2
SET GRAVITY ( 0.0, -10.0 )

CYC 50000
#删除10000米以上的颗粒，可实现剥蚀
DEL RANGE y (11000.0, 25000.0)
CYC 10000
EXP initxyr.dat RANGE GROUP ball_rand
SAV initxyr.sav
```

(续下页)

(接上页)

```

##### 设置颜色 #####
prop color lg RANGE GROUP ball_rand
prop color mg RANGE x 101.0 59999.0 y 261.0 1000.0
prop color mg RANGE x 101.0 59999.0 y 2000.0 3000.0
prop color mg RANGE x 101.0 59999.0 y 4000.0 5000.0
prop color mg RANGE x 101.0 59999.0 y 6000.0 7000.0
prop color mg RANGE x 101.0 59999.0 y 8000.0 9000.0
prop color mg RANGE x 101.0 59999.0 y 10000.0 11000.0
PROP FRIC 0.3 RANGE GROUP bom_wall1
prop FRIC 0.3 RANGE GROUP bom_wall2
#设置粘结
PROP ebmod 2e8 gbmod 2e8 tstrength 2e7 sstrength 4e7 fric 0.3 RANGE GROUP ball_rand
#开始伸展
INI XV 2.0 RANGE GROUP bom_wall2
WALL ID 1 XV 2.0
IMPLE WALL ID 1 XMOVE 2000.0 SAVE 2000.0 PRINT 1000.0 PS 1000.0

##### 沉积1 #####
gen NUM 200000, rad discrete 60.0 80.0 , x( 10000.0, 42000.0) y (10000.0, 14000.0 ) GROUP_
↳ballsed1
PROP COLOR blue DEN 2.5e3 FRIC 0.3 SHEAR 2.9e9 POISS 0.2 DAMP 0.0 HERTZ RANGE GROUP ballsed1
#新沉积的颗粒粘结(胶结)在一起
#prop ebmod 2e8 gbmod 2e8 tstrength 2e7 sstrength 4e7 fric 0.3 range group ballsed1
INI XV 0.0 RANGE GROUP bom_wall2
WALL ID 1 XV 0.0
set ps 1000
set print 1000
CYC 5000
DEL RANGE y 11000.0 16000.0
CYC 2000
#开始伸展
INI XV 2.0 RANGE GROUP bom_wall2
wall id 1 xv 2.0
IMPLE WALL ID 1 XMOVE 2000.0 SAVE 2000.0 PRINT 1000.0 PS 1000.0

##### 沉积2 #####
gen NUM 200000, rad discrete 60.0 80.0 , x( 10000.0, 44000.0) y (10000.0, 14000.0 ), GROUP_
↳ballsed2
PROP color red den 2.5e3, fric 0.3, shear 2.9e9, poiss 0.2, damp 0.0, hertz range group ballsed2
INI XV 0.0 RANGE GROUP bom_wall2
WALL ID 1 XV 0.0
set ps 1000
set print 1000
CYC 5000
DEL RANGE y 11000.0 16000.0
CYC 2000

#开始伸展
INI XV 2.0 RANGE GROUP bom_wall2
wall id 1 xv 2.000
IMPLE WALL ID 1 XMOVE 2000.0 SAVE 2000.0 PRINT 1000.0 PS 1000.0

##### 沉积3 #####
gen NUM 200000, rad discrete 60.0 80.0 , x( 10000.0, 46000.0) y (10000.0, 14000.0 ), GROUP_

```

(续下页)

(接上页)

```

↔ballsed3
PROP color mg den 2.5e3, fric 0.3, shear 2.9e9, poiss 0.2, damp 0.0, hertz range group ballsed3
INI XV 0.0 RANGE GROUP bom_wall2
WALL ID 1 XV 0.0
set ps 1000
set print 1000
CYC 5000
DEL RANGE y 11000.0 16000.0
CYC 2000

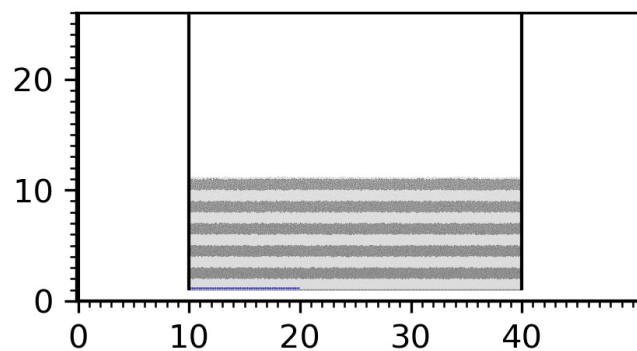
#开始伸展
INI XV 2.0 RANGE GROUP bom_wall2
wall id 1 xv 2.000
IMPLE WALL ID 1 XMOVE 2000.0 SAVE 2000.0 PRINT 1000.0 PS 1000.0

##### 沉积4 #####
gen NUM 200000, rad discrete 60.0 80.0 , x( 10000.0, 48000.0) y (10000.0, 14000.0 ), GROUP
↔ballsed4
PROP color gb den 2.5e3, fric 0.3, shear 2.9e9, poiss 0.2, damp 0.0, hertz range group ballsed4
INI XV 0.0 RANGE GROUP bom_wall2
WALL ID 1 XV 0.0
set ps 1000
set print 1000
CYC 5000
DEL RANGE y 11000.0 16000.0
CYC 2000

#结束计算
STOP

```

计算结束后, 将得到以下结果:



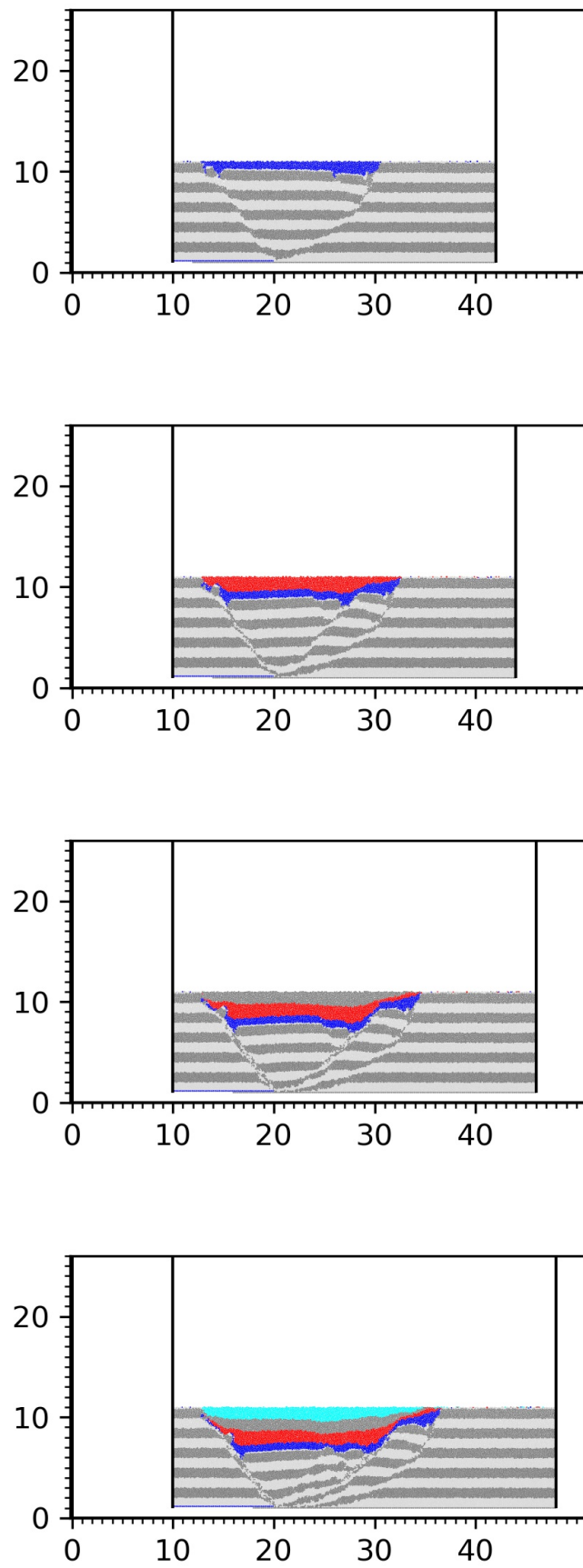


图 5.6: 单位 (km)

参考文献

1. 李长圣, 尹宏伟*, 张佳星, 汪伟. 琼东南盆地基底断层性质对凹陷沉积模式的影响: 基于离散元数值模拟的认识. **2018 中国地球科学联合学术年会论文集**, 北京, 2018.

5.7 与物理模拟对比实验

备注: ~10 万颗粒, ~13 万时步, 16 核并行 (Intel Xeon E5-2650), 耗时 ~3 小时完成计算。

构建了一个典型离散元数值模拟 (DEM) 试验, 对比与构造物理模拟 (AM) 和差异与相似点。

DEM 与 AM 模拟结果较为一致, 基本反映了 AM 中石英砂的变形行为。AM 中材料选取有限 (如硅胶、粘土、微玻璃珠等), 而 DEM 材料选取范围大, 但其模拟结果依赖参数选取。AM 和 DEM 作为两种独立的方法, 有很好的互补性 [Li2021]。

数值模拟分两步进行:

1. `gen.py` 生成颗粒, 沉积, 构建初始模型。
2. `push.py` 给定颜色、粘结参数, 挤压。

表 1 颗粒微观参数表. [Li2021]

$d(mm)$	$k(N \cdot m^{-1})$	$(kg \cdot m^{-2})$	$g(m \cdot s^{-2})$	f	$(N \cdot s \cdot m^{-1})$	$(m \cdot s^{-1})$
0.6	7.5e3	1.3e3	10.0	0.4	0.3	0.04

The particle packing consists of four particle sizes, with diameters and quantity ratio of 0.2 mm, 0.4 mm, 0.5 mm, and 0.6 mm and 2:2:1:1, respectively. d , largest particle diameter. ρ , particle density. g , gravitational acceleration. f , safety factor of the time step. k , stiffness of the contact. μ , friction coefficient. η , dynamic viscosity. v , velocity of the mobile wall.

`gen.py`

```
#####
# title: 离散元数值模拟与构造物理模拟对比试验:1 沉积
# date: 2021-05-16
# authors: 李长圣
# E-mail: sheng0619@163.com
# ref: Li et al. (2021) Calibration of the discrete element method and modelling of shortening
↳ experiments. Front. Earth Sci. doi: 10.3389/feart.2021.636512
# more info, see www.geovbox.com
#####
start
```

(续下页)

(接上页)

```

set disk hex
BOX left 0.5e-3 right 615.0e-3 bottom 0.5e-3 height 110.0e-3 kn=1.5e4 ks=1.5e4 fric 0.3

wall id 4, nodes ( 5e-3 158.0e-3) ( 5.0e-3 5.0e-3), kn= 1.5e4 ks= 1.5e4 fric 0.0
wall id 5, nodes ( 5e-3 5.0e-3) (605.0e-3 5.0e-3), kn= 1.5e4 ks= 1.5e4 fric 0.0
wall id 6, nodes ( 605.0e-3 5.0e-3) (605.0e-3 158.0e-3), kn= 1.5e4 ks= 1.5e4 fric 0.0

gen grid idmin 0 rad discrete 0.1e-3 0.1e-3 0.2e-3 0.2e-3 0.25e-3 0.3e-3, x (5.0e-3, 605.0e-3), y
↳(5.0e-3, 155.0e-3), GROUP ball_rand

PROP color red den 1.3e3, fric 0.0, kn 1.5e4, ks 1.5e4 damp 0.4
FIX spin

SET frac 0.4
SET GRAVITY ( 0.0, -10.0 )
SET stepbar 1000
SET save 20000
SET print 20000
SET ps 20000
HIST ID 1 INTERVAL 1000 , kinetic
HIST ID 2 INTERVAL 1000 , step
PLOT hist 2 1

CYC 60000
DEL range x (4.0e-3, 606.0e-3), y (30.0e-3, 1.0),
CYC 20000
#save 2del.sav
EXP ini_xyr.dat
STOP

```

push.py

```

#####
# title: 离散元数值模拟与构造物理模拟对比试验:2 挤压
# date: 2021-05-16
# authors: 李长圣
# E-mail: sheng0619@163.com
# ref: Li et al. (2021) Calibration of the discrete element method and modelling of shortening
↳experiments. Front. Earth Sci. doi: 10.3389/feart.2021.636512
# more info, see www.geovbox.com
#####
LOAD ini_xyr.dat
SET disk hex
BOX left 0.5e-3 right 615.0e-3 bottom 0.5e-3 height 110.0e-3 kn=1.5e4 ks=1.5e4 fric 0.3

WALL id 4, nodes ( 5e-3 110.0e-3) ( 5.0e-3 5.0e-3), kn= 1.5e4 ks= 1.5e4 fric 0.0
WALL id 5, nodes ( 5e-3 5.0e-3) (605.0e-3 5.0e-3), kn= 1.5e4 ks= 1.5e4 fric 0.0
WALL id 6, nodes ( 605.0e-3 5.0e-3) (605.0e-3 110.0e-3), kn= 1.5e4 ks= 1.5e4 fric 0.0

PROP color red den 1.3e3, fric 0.0, kn 1.5e4, ks 1.5e4 damp 0.0
FIX spin

SET frac 0.4

```

(续下页)

(接上页)

```

SET GRAVITY ( 0.0, -10.0 )
SET stepbar 10000
HIST ID 1 INTERVAL 1000 , kinetic
HIST ID 2 INTERVAL 1000 , step
PLOT 2 1

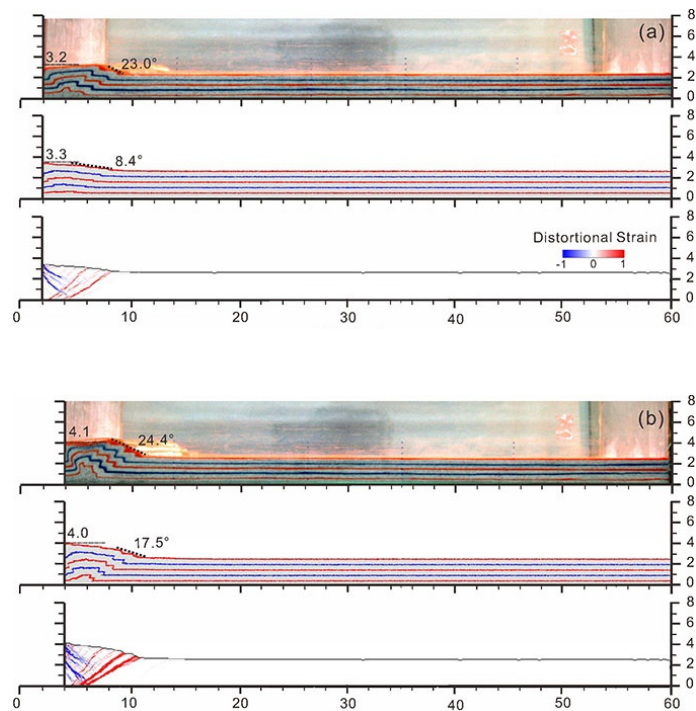
SET damp lsm 0.04
PROP fric 0.30
PROP color lg
PROP color red , range x 0.0 615.0e-3 y 9.0e-3 10.0e-3
PROP color blue, range x 0.0 615.0e-3 y 14.0e-3 15.0e-3
PROP color red , range x 0.0 615.0e-3 y 19.0e-3 20.0e-3
PROP color blue, range x 0.0 615.0e-3 y 24.0e-3 25.0e-3
PROP color red , range x 0.0 615.0e-3 y 29.0e-3 30.0e-3

wall id 4 fric 0.30 xv 40e-3
wall id 5 fric 0.30
wall id 6 fric 0.30

CYC 1
imple wall id 4 xmove 160e-3 save 20e-3 print 10e-3 ps 10e-3 vtk 10e-3
stop

```

计算结束后, 将得到以下结果:



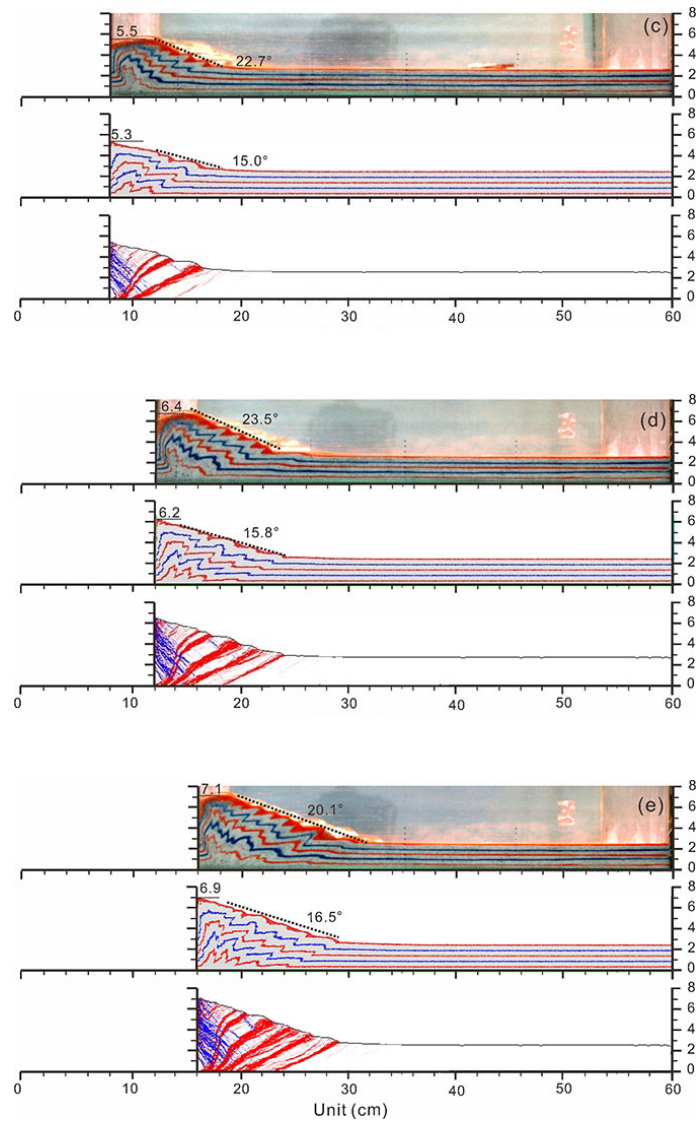


图 5.7: 单位 (cm)

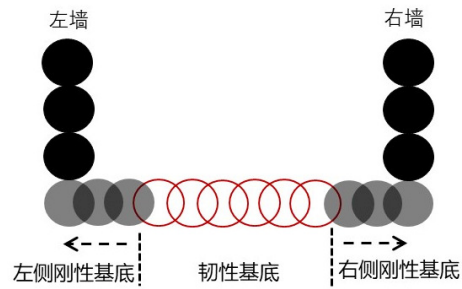
参考文献

5.8 韧性基底伸展构造

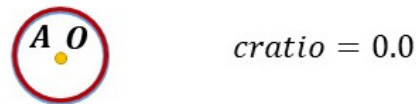
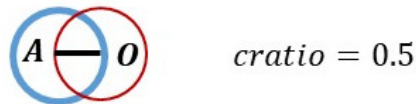
这里是一个韧性基底伸展构造设置的实例，基底由左右两个刚性墙和中间的韧性墙组成。

粘结时，设置临时半径缩放系数等于初始叠合率，避免出现跨颗粒粘结情况发生，详细描述见发明专利：

李长圣, 尹宏伟, 吴珍云, 等. 一种基于离散元的裂谷盆地伸展过程模拟方法 [P]. 江西省: CN111008472B, 2023-11-21.

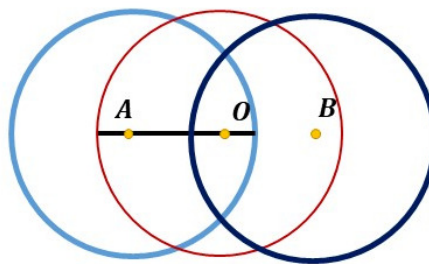


- 初始叠合率 $cratio = |AO| / (r_A + r_O)$, 即圆心距离 ($|AO|$) 与平衡距离 ($r_A + r_O$) 的比值

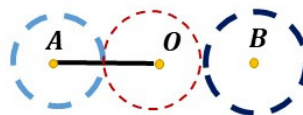


$$cratio = |AO| / (r_A + r_O)$$

- 临时半径缩放系数 $r_{ext} = r_{tmp} / r_{old}$



(a) $cratio = 0.4$



(b) $r_{ext} = 0.4$

`extens_ductile_patent.py` 中内容如下:

```
#2024-04-04
#建议使用notepad++查看本文件
#LI ChangSheng @ NanJing University
```

(续下页)

(接上页)

```

#E-mail: sheng0619@163.com
#more info, see www.geovbox.com
#####
# title: 韧性基底伸展构造实例
#详细解释见发明专利: 李长圣,尹宏伟,吴珍云,等. 一种基于离散元的裂谷盆地伸展过程模拟方法[P].
↪江西省: CN111008472B,2023-11-21.
#本示例初始叠合率cratio=临时半径缩放系数rest=0.4, 取值范围0.0-1.0
#括号内参数可根据模型大小及个人需要修改
#脚本命令不区分大小写
#####
START
SET disk 0
BOX left 1.0e-3 right 40000.0 bottom 1.0e-3 height 12000.0 kn=4e10 ks=4e10 fric 0.30

#####
↪#
#初始叠合率 cratio=|AO|/(rA+rO),即圆心距离(|AO|)与平衡距离(rA+rO)的比值
GLINE p1 ( 10000.0 , 1000.0 ) p2 ( 25000.0 , 1000.0 ), rad 80.0 cratio 0.4 color=red GROUP wall_
↪bom
#####
↪#
GLINE p1 ( 10000.0 , 1160.0 ) p2 ( 10000.0 , 11000.0 ), rad 80.0 color=red GROUP wall_lef
GLINE p1 ( 25000.0 , 1160.0 ) p2 ( 25000.0 , 11000.0 ), rad 80.0 color=red GROUP wall_rig

FIX x y spin RANGE group wall_bom
FIX x y spin RANGE group wall_lef
FIX x y spin RANGE group wall_rig

GEN NUM 200000, rad discrete 60.0 80.0, x 10000.0, 25000.0, y 1000.0, 8000.0, color black GROUP
↪ball_rand
PROP den 2.5e3, fric 0.0, shear 2.9e9, poiss 0.2, damp 0.4, hertz
DRAW INTERVAL 200, bfill wall
SET STEPBAR 1000
SET print 5000
SET ps 5000
SET DT 5e-2,
SET GRAVITY ( 0.0, -10.0 )

CYC 10000
DEL RANGE x 10001.0, 24999.0, y 4000.0, 15000.0
CYC 5000
EXP initxyr.dat RANGE group ball_rand
SAV initxyr.sav

PROP color mg RANGE group ball_rand
PROP color lg RANGE x 10001.0 24999.0 y 2000.0 2200.0
PROP color lg RANGE x 10001.0 24999.0 y 3000.0 3200.0
PROP color lg RANGE x 10001.0 24999.0 y 4000.0 4200.0

PROP fric 0.3 RANGE GROUP wall_bom
PROP fric 0.3 RANGE GROUP wall_lef
PROP fric 0.3 RANGE GROUP wall_rig

PROP ebmod 2e8 gbmod 2e8 tstrength 2e7 sstrength 4e7 fric 0.3 RANGE group ball_rand

```

(续下页)

(接上页)

```

PROP group wall_mov0 color blue RANGE x 9000.0 13000.0 y 999.0 1001.0
PROP group wall_mov1 color blue RANGE x 22000.0 31000.0 y 999.0 1001.0
#####
↪#
#临时半径缩放系数 $rext=r_{tmp}/r_{old}$ 
PROP tolerate rext 0.4 ebmod 2e8 gbmod 2e8 tstrength 2e100 sstrength 4e7 fric 0.3 RANGE group wall_
↪bom and group wall_mov0
PROP tolerate rext 0.4 ebmod 2e8 gbmod 2e8 tstrength 2e100 sstrength 4e7 fric 0.3 RANGE group wall_
↪bom and group wall_mov1
#####
↪#

#释放中间部分的墙体，以便其可以在左右墙体拉伸下，可以自由伸展开
FREE x RANGE group wall_bom

#墙体wall_rig和wall_mov1向右移动
INI xv 2.0, RANGE group wall_rig
INI xv 2.0, RANGE group wall_mov1
#墙体wall_lef和wall_mov0向左移动
INI xv -2.0, RANGE group wall_lef
INI xv -2.0, RANGE group wall_mov0
#设置辅助墙体，用于设置伸展距离，这里设置墙体id=1向右移动1000m
WALL id 1 nodes ( 26000.0 1080.0 ) ( 26000.0 5000.0 ), kn=0e3 ks=0e3 fric=0.0 color=black
WALL id 1 xv 2.0
IMPLE wall id 1 xmove 1000.0 print 500.0 ps 500.0

#沉积1
GEN NUM 200000, rad discrete 60.0 80.0 , x( 9000.0,26000.0) y ( 4000.0, 6000.0 ), GROUP ballsed2
PROP color blue den 2.5e3, fric 0.3, shear 2.9e9, poiss 0.2, damp 0.4, hertz RANGE group ballsed2
#墙体停止移动
WALL id 1 xv 0.0
INI xv 0.0, RANGE group wall_lef
INI xv 0.0, RANGE group wall_rig
INI xv 0.0, RANGE group wall_mov1
INI xv 0.0, RANGE group wall_mov0
SET ps 1000
SET print 1000
CYC 5000
DEL RANGE x 9001.0 25999.0 y 4000.0 10000.0
CYC 1000

#伸展-墙体继续移动
wall id 1 xv 2.0
INI xv 2.0, RANGE group wall_rig
INI xv -2.0, RANGE group wall_lef
INI xv 2.0, RANGE group wall_mov1
INI xv -2.0, RANGE group wall_mov0
IMPLE WALL id 1 xmove 1000.0 save 500.0 print 500.0 ps 500.0

#沉积2
gen NUM 200000, rad discrete 60.0 80.0 , x(8000.0,27000.0) y (4000.0, 6000.0 ), GROUP ballsed3
PROP color gb den 2.5e3, fric 0.3, shear 2.9e9, poiss 0.2, damp 0.4, hertz RANGE group ballsed3
#墙体停止移动
WALL id 1 xv 0.0

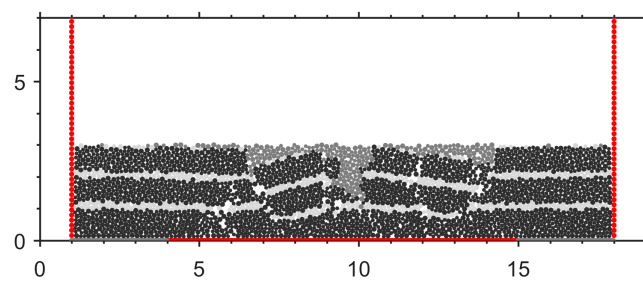
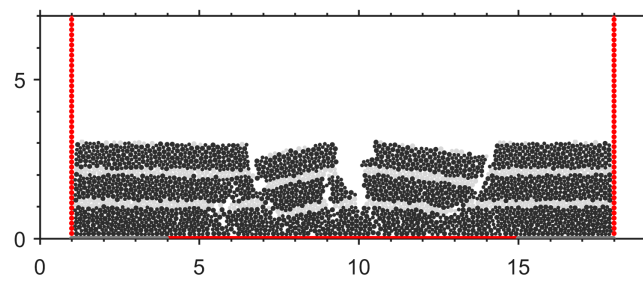
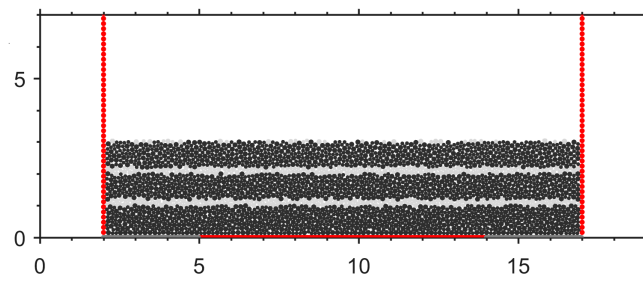
```

(续下页)

(接上页)

```
ini xv 0.0, RANGE group wall_lef
ini xv 0.0, RANGE group wall_rig
ini xv 0.0, RANGE group wall_mov1
ini xv 0.0, RANGE group wall_mov0
set ps 1000
set print 1000
CYC 5000
DEL RANGE x 8001.0 26999.0 y 4000.0 10000.0
CYC 1000
sav sed5km.sav
stop
```

计算结束后,将得到以下结果:



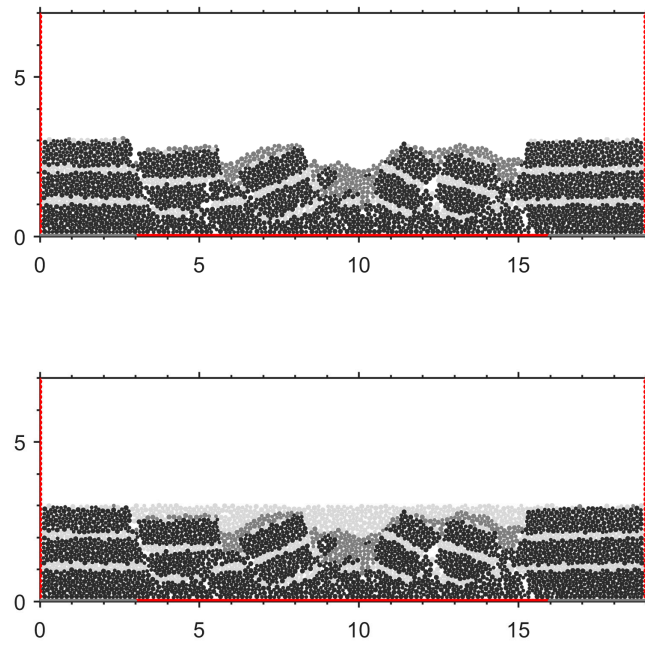


图 5.8: 单位 (km)

第 6 章 应力应变

6.1 配置环境变量

绘制应力应变图, 需要引用如下文献:

1. Li C, Yin H, et al. (2021) Calibration of the Discrete Element Method and Modeling of Shortening Experiments. *Front. Earth Sci.* 9:636512.
2. 李长圣 (2019) 基于离散元的褶皱冲断带构造变形定量分析与模拟. 博士学位. 南京大学. Li, C. (2019). Quantitative Analysis and Simulation of Structural Deformation in the Fold and Thrust Belt Based on Discrete Element Method. Doctor Thesis. NanJing University.(in Chinese with English abstract) [推荐下载](#) [最新修订版](#) [提取码 zdem](#)
3. Morgan JK (2015) Effects of cohesion on the structural and mechanical evolution of fold and thrust belts and contractional wedges: Discrete element simulations. *Journal of Geophysical Research: Solid Earth* 120:3870-3896.

配置 zdemss 的环境变量, 将如下内容写入 `~/.bashrc` 的文件尾部。

```
#gmt
source /public1/soft/modules/module.sh
source /public1/soft/other/module_GMT.sh
module load GMT_5.4.5

#zdemss
export PATH=/public1/home/sch3726/bin:$PATH #T6分区
#export PATH=/public1/home/sc80502/bin:$PATH #T分区
```

备注: 不会配置 zdemss 环境变量的小白用户可参考 [bilibili 视频之配置 zdemss 的环境变量](#)

6.2 无沉积剥蚀

如果没有沉积和剥蚀过程， `sbatch jobs.sh` 提交， 应力应变将输出到 `ex5_detachment/data/ss`。

这里采用 构造模拟中的[滑脱层](#) 示例

目录结构：

```
| -- ex5_detachment
|   |-- detachment.py
|   |-- init_xyr.dat
|   |-- job.sh
```

- `detachment.py` 见[滑脱层](#)

警告： 注意示例 `detachment.py` 中没有使用 `GEN` 命令生成颗粒，而是采用的 `LOAD init_xyr.dat` 导入初始模型。如果使用了 `GEN`，那么你可能引入了沉积过程，需要参考[有沉积剥蚀](#) 去除沉积过程生成的 `all_*.dat`。

- `init_xyr.dat` 见[滑脱层](#)
- `job.sh` 内容如下：

```
#!/bin/bash
#SBATCH --job-name=core12
#SBATCH --partition=v6_384
#SBATCH -n 1
#SBATCH -c 12
#SBATCH -t 1440
#SBATCH --output=%j.out
#SBATCH --error=%j.err

#zdem environment variable
source /public1/soft/modules/module.sh
source /public1/soft/other/module_zdem.sh
module load zdem2.0

#GMT environment variable
source /public1/soft/modules/module.sh
source /public1/soft/other/module_GMT.sh
module load GMT_5.4.5

#zdemss environment variable
export PATH=/public1/home/sc80502/bin:$PATH

time srun -n 1 zdem detachment.py
time srun -n 1 zdem2jpg --dir=./data
time srun -n 1 zdem -j 12 -s ./data
time srun -n 1 zdemss --dir ./data
```

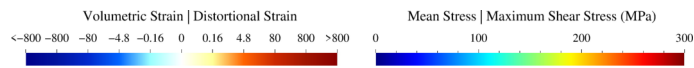
其中,

- `time` 记录该行命令的运行时间, 输出到 `%j.err`
- `zdem2jpg --dir=./data` 生成 jpg, 详解见 [zdem2jpg](#) 。
- `zdem -j 12 -s ./data` 计算应力应变, 详解见 [zdem](#) 。
- `zdemss --dir ./data` 绘制应力应变, 详解见 [zdemss](#) 。

警告: 本示例采用 Slurm 作业调度系统提交计算。注意 `job.sh` 的行结尾符需采用 Unix 的 `\n` 格式, 否则将产生如下错误信息:

```
sbatch: error: Batch script contains DOS line breaks (\r\n)
sbatch: error: instead of expected UNIX line breaks (\n).
```

`job.sh` 中命令解析及如何修改 `job.sh` 的行结尾符见 [第 00 课并行超算云上使用 ZDEM 的 10:15](#)



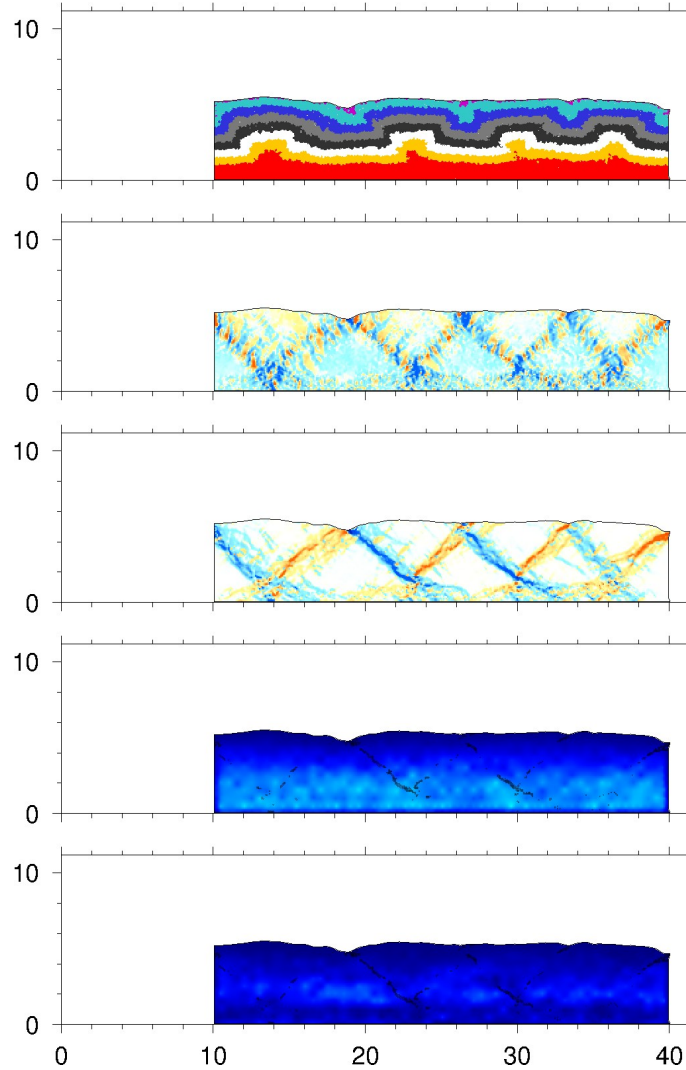


图 6.1: 处理结果示意图。从上到下, 分别为: 变形图, 体积应变, 变形应变, 平均应力, 最大剪切应力。

6.3 有沉积剥蚀

6.3.1 流程解析

警告: 此节仅为了帮助用户理解应力应变计算过程, 用户仍需要 Slurm 作业调度系统 (*Slurm* 作业调度系统) 提交计算, 具体方法见实例。

初始目录结构:

```
|-- ex_strain_stress
|-- push_add_del.py
```

push_add_del.py 见实例

如果有沉积剥蚀过程, 需要仔细选择沉积剥蚀的临界文件, 详细的流程如下:

所有操作均在 `ex_strain_stress` 目录下操作

1. `zdem push_add_del.py`

计算完成, 将生成 `./data` 文件夹。目录结构如下:

```
|-- ex_strain_stress
    |-- push_add_del.py
    |-- data
        |-- all_0000000000_ini.dat
        |-- ...
        |-- all_0000005000.dat
        |-- all_0000058000_ini.dat
        |-- all_0000108000.dat
```

2. `zdem2jpg --dir=./data`

生成 jpg, 生成计算过程图. 注意: 这里, 只需指定 `--dir`, 不加其它任何参数. 目录结构如下

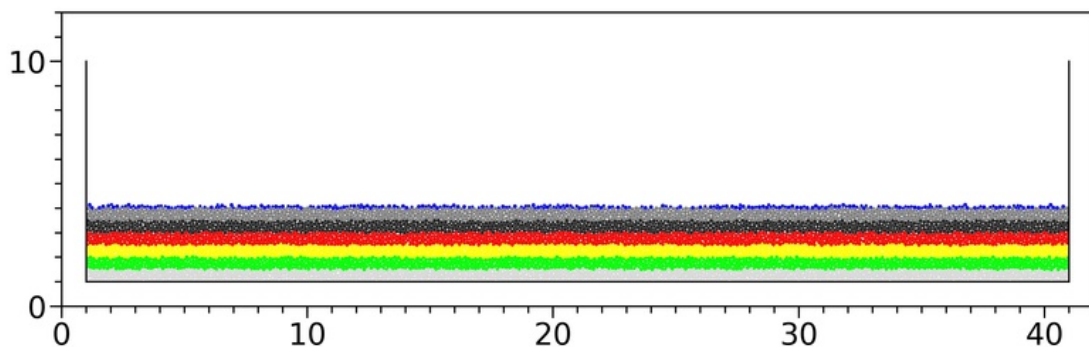
```
|-- ex_strain_stress
    |-- push_add_del.py
    |-- data
        |-- all_0000000000_ini.dat
        |-- all_0000000000_ini.jpg
        |-- ...
        |-- all_0000005000.dat
        |-- all_0000005000.jpg
        |-- all_0000058000_ini.dat
        |-- all_0000058000_ini.jpg
        |-- all_0000108000.dat
        |-- all_0000108000.jpg
```

3. 新建 `datass` 文件夹, 根据 jpg 挑选需要计算应力应变的 .dat 文件, 复制到 `datass` 文件夹中。

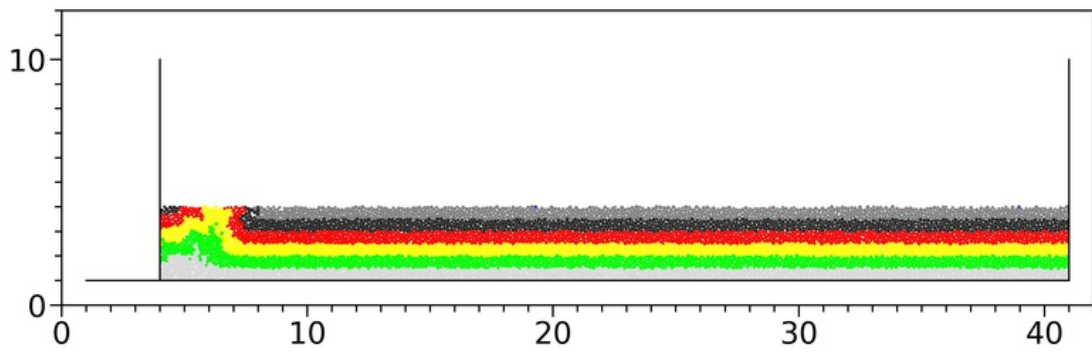
这里, 挑选原则如下:

- 刚沉积完, 并给定颜色, 准备挤压的初始模型, 必须。

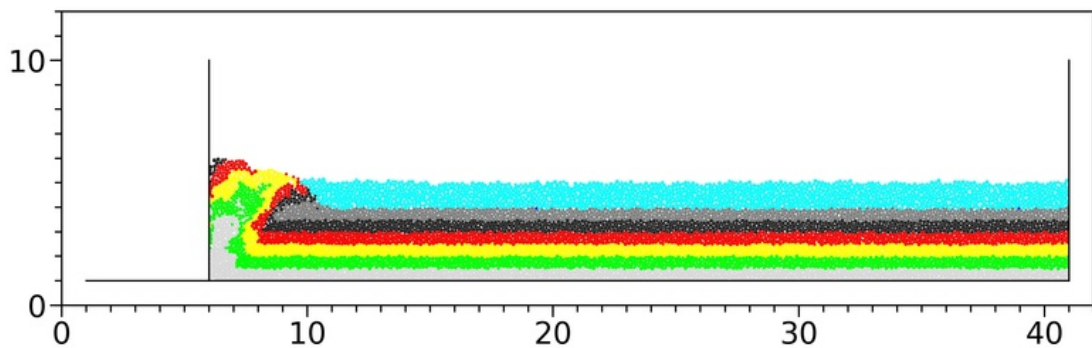
`all_0000006000_ini.dat`



- 刚剥蚀完, 准备挤压的模型, 必须。 `all_0000036000_ini.dat`



- 沉积稳定, 准备挤压前的模型。all_0000058000_ini.dat



- 沉积过程不要, 其它的可酌情选取。

目录结构如下

```
|-- ex_strain_stress
  |-- push_add_del.py
  |-- data
    |-- all_0000000000_ini.dat
    |-- all_0000000000_ini.jpg
    |-- ...
    |-- all_0000005000.dat
    |-- all_0000005000.jpg
    |-- all_0000058000_ini.dat
    |-- all_0000058000_ini.jpg
    |-- all_0000108000.dat
    |-- all_0000108000.jpg
  |-- datass
    |-- all_0000006000_ini.dat
    |-- all_0000026000.dat
    |-- all_0000036000_ini.dat
    |-- all_0000056000.dat
    |-- all_0000058000_ini.dat
    |-- all_0000078000.dat
    |-- all_0000108000.dat
```

4.zdem --xmove -1000.0 --ymove -1000.0 --addball --delball -s ./datass

计算应力和应变。基于步骤 2, 我们知道 --xmove --ymove 应该设置为 -1000.0

, 并且有沉积 `--addball` 和剥蚀 `--delball` 过程。

备注: `zdem` 命令详解见 [zdem](#), 或者输入 `zdem -h` 查看帮助文档。

计算完成, 将生成 `./datass/ss/data/*.out`, 供 GMT 绘图用。目录结构如下

```
|-- ex_strain_stress
  |-- push_add_del.py
  |-- data
    |-- all_0000000000_ini.dat
    |-- all_0000000000_ini.jpg
    |-- ...
    |-- all_0000005000.dat
    |-- all_0000005000.jpg
    |-- all_0000058000_ini.dat
    |-- all_0000058000_ini.jpg
    |-- all_0000108000.dat
    |-- all_0000108000.jpg
  |-- datass
    |-- ss
      |-- data
        |-- *.out
        |-- all_0000006000_ini.dat
        |-- all_0000026000.dat
        |-- all_0000036000_ini.dat
        |-- all_0000056000.dat
        |-- all_0000058000_ini.dat
        |-- all_0000078000.dat
        |-- all_0000108000.dat
```

5. `zdemss --dir ./datass --addball ON --delball ON --xmax 40.0 --ymax 10.0 --maxstress 250.0` 使用 GMT 绘制应力应变。`zdemss` 将会读取 `zdem` 生成的应力应变数据 `./datass/ss/data`, 生成应力应变云图到 `./datass/ss/*.jpg`。设置 x 轴最大值 40.0 km, 设置 y 轴最大值 10.0 km, 设置颜色条应力最大值 250 MPa。

备注: `zdemss` 命令详解见 [zdemss](#), 或者输入 `zdemss -h` 查看帮助文档。

运行完成之后, 目录结构:

```
|-- ex_strain_stress
  |-- push_add_del.py
  |-- data
    |-- all_0000000000_ini.dat
    |-- ...
    |-- all_0000005000.dat
    |-- all_0000058000_ini.dat
    |-- all_0000108000.dat
  |-- datass
```

(续下页)

(接上页)

```

|-- ss
  |-- data
    |-- *.out
  |-- ps
    |-- *.ps
  |-- tmp
    |-- *.grd
  |-- *.jpg
|-- all_0000006000_ini.dat
|-- all_0000026000.dat

```

- ./datass/ss/data/*.out *zdem* 输出的颗粒位置和 (假) 应力的原始数据
- ./datass/ss/data2ps/*.out (真) 应力应变数据, 用于绘制应力应变图 (ps), 用户可以用于定量分析
- ./datass/ss/ps/*.ps 输出的应力应变图 (矢量图)
- ./datass/ss/Tmp/*.grd 计算应力应变产生的中间数据
- ./datass/ss/*.jpg 输出的应力应变图 (位图)

6.3.2 实例

1. sbatch jobs1.sh 提交, 将完成[流程解析 1](#) 和 2
2. **此步最关键!** 新建 datass 文件夹, 将需要处理的 dat 复制到 datass 文件夹, 完成[流程解析 3](#)。完成此步后, 目录结构:

```

|-- ex_strain_stress
  |-- job1.sh
  |-- job2.sh
  |-- push_add_del.py
  |-- data
    |-- all_0000000000_ini.dat
    |-- all_0000001000.dat
    |-- ...
    |-- all_0000005000.dat
    |-- all_0000005000_ini.dat
    |-- all_0000006000.dat
    |-- all_0000006000_ini.dat
    |-- all_0000016000.dat
    |-- all_0000026000.dat
    |-- all_0000036000.dat
    |-- all_0000036000_ini.dat
    |-- all_0000046000.dat
    |-- all_0000056000.dat
    |-- all_0000056000_ini.dat
    |-- all_0000056100.dat
    |-- ...
    |-- all_0000057900.dat
    |-- all_0000058000.dat
    |-- all_0000058000_ini.dat
    |-- all_0000068000.dat
    |-- all_0000078000.dat
    |-- all_0000088000.dat

```

(续下页)

(接上页)

```

|-- all_0000098000.dat
|-- all_0000108000.dat
|-- datass
|-- all_0000006000_ini.dat
|-- all_0000026000.dat
|-- all_0000036000_ini.dat
|-- all_0000056000.dat
|-- all_0000058000_ini.dat
|-- all_0000078000.dat
|-- all_0000108000.dat

```

3. `sbatch jobs2.sh` 提交, 将完成[流程解析](#) 4 和 5。等待计算完成, 生成的应力应变图见 `./datass/ss/*jpg`。

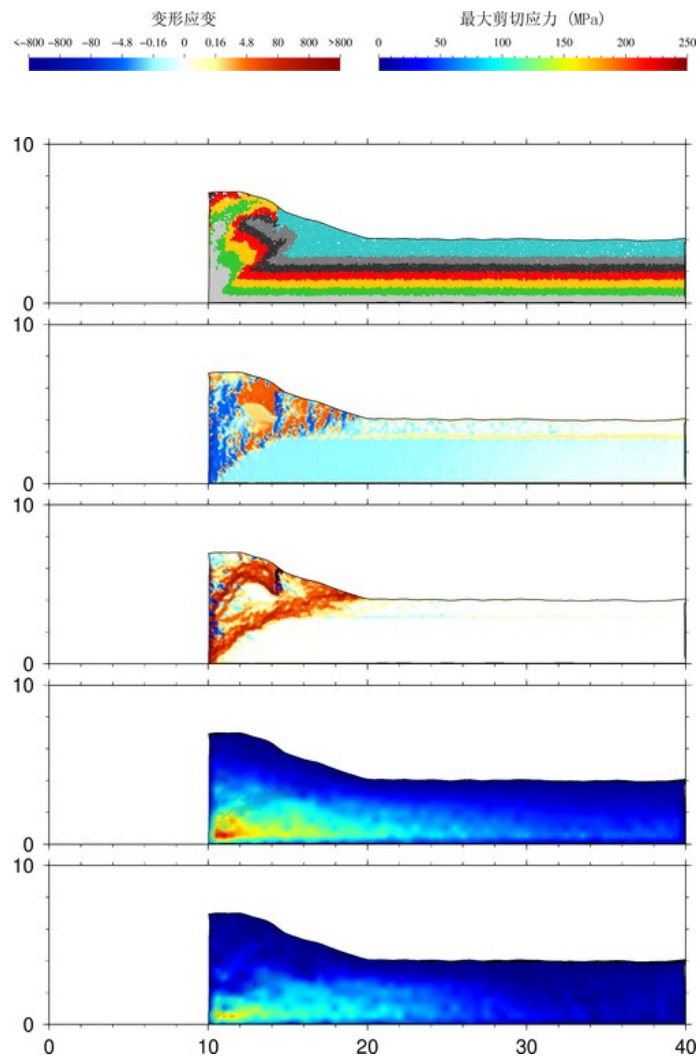


图 6.2: 处理结果示意图

`job1.sh job2.sh push_add_del.py` 文件内容:

`job1.sh`

```
#!/bin/bash
#SBATCH --job-name=core12
#SBATCH --partition=v6_384
#SBATCH -n 1
#SBATCH -c 12
#SBATCH -t 1440
#SBATCH --output=%j.out
#SBATCH --error=%j.err

source /public1/soft/modules/module.sh
source /public1/soft/other/module_zdem.sh
module load zdem2.0

source /public1/soft/modules/module.sh
source /public1/soft/other/module_GMT.sh
module load GMT_5.4.5

export PATH=/public1/home/sc80502/bin:$PATH

time srun -n 1 zdem push_add_del.py
time srun -n 1 zdem2jpg --dir=./data
```

job2.sh

```
#!/bin/bash
#SBATCH --job-name=core12
#SBATCH --partition=v6_384
#SBATCH -n 1
#SBATCH -c 12
#SBATCH -t 1440
#SBATCH --output=%j.out
#SBATCH --error=%j.err

source /public1/soft/modules/module.sh
source /public1/soft/other/module_zdem.sh
module load zdem2.0

source /public1/soft/modules/module.sh
source /public1/soft/other/module_GMT.sh
module load GMT_5.4.5

export PATH=/public1/home/sc80502/bin:$PATH

time srun -n 1 zdem --xmove -1000.0 --ymove -1000.0 -j 12 --addball --delball -s ./datass
time srun -n 1 zdemss --dir ./datass --xmax 40.0 --ymax 10.0 --maxstress 250.0 --addball ON --
↵delball ON
```

push_add_del.py

```
#####
# title: 一个实例学会VBOX 加入剥蚀 沉积 演示应力应变处理过程
# date: 2020-06-28
# authors: 李长圣
```

(续下页)

(接上页)

```

# E-mail: sheng0619@163.com
# www.geovbo.com
#####
#程序初始化
START
#颗粒设为球，计算颗粒体积用 $4/3\pi r^3$ 计算
set disk off
#设置研究范围
BOX left 0.0 right 42000.0 bottom 0.0 height 12000.0 kn=0e10 ks=0e10 fric 0.00
#设置挡板墙，这里模型采用hertz接触模型，挡板墙的kn ks无效，计算时取颗粒的参数
WALL ID 0, NODES ( 1000.0 , 1000.0 ) ( 41000.0 , 1000.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR_
↪black
WALL ID 1, NODES ( 1000.0 , 10000.0 ) ( 1000.0 , 1000.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR_
↪blue
WALL ID 2, NODES ( 41000.0 , 1000.0 ) ( 41000.0 , 10000.0 ), kn=0e10 ks=0e10 fric 0.0 COLOR red
#在矩形范围内生成颗粒
GEN NUM 100000.0 rad discrete 60.0 80.0, x ( 1000.0, 41000.0), y ( 1000.0, 10000.0), COLOR black_
↪GROUP ball_rand
#设置颗粒的微观参数
PROP DENSITY 2.5e3, fric 0.0, shear 2.9e9, poiss 0.2, damp 0.4, hertz
#设置时间步及重力加速度
SET DT 5e-2, GRAVITY 0.0, -10.0
#设置每1000步保存一次ps格式的计算结果
SET ps 1000
#设置每1000步保存一次dat格式的计算结果
SET print 1000
#沉积，计算5000步
CYC 5000
#删除4000米以上的颗粒
DEL RANGE y 4000.0 999000.0
#平衡，计算1000步
CYC 1000
#输出包含颗粒的[x y r]信息的初始模型 init_xyr.dat
EXP init_xyr.dat

#设置bond粘结，使颗粒具有粘聚力
PROP ebmod 2e8 gbmod 2e8 tstrength 2e7 sstrength 4e7 fric 0.3
#给地层赋上颜色
PROP COLOR lg range y 1000.0 1500.0
PROP COLOR green range y 1500.0 2000.0
PROP COLOR yellow range y 2000.0 2500.0
PROP COLOR red range y 2500.0 3000.0
PROP COLOR black range y 3000.0 3500.0
PROP COLOR mg range y 3500.0 4000.0
PROP COLOR blue range y 4000.0 4500.0
PROP COLOR gb range y 4500.0 5000.0
PROP COLOR violet range y 5000.0 5500.0

#设置挡板墙摩擦系数
WALL id 0 fric 0.3
WALL id 1 fric 0.3
WALL id 2 fric 0.3
#设置墙的挤压速度 x方向速度为2.0
WALL id 1 xv 2.0

```

(续下页)

(接上页)

```

#设置墙的挤压量x方向推进3000.0, 每挤压2000.0保存一次计算结果
IMPLE wall id 1 xmove 3000.0 save 2000.0 print 1000.0 ps 1000.0

##### 剥蚀 #####
# 删除4000米以上的颗粒
DEL RANGE y 4000.0 999000.0
#####

#设置墙的挤压量x方向推进3000.0, 每挤压2000.0保存一次计算结果
IMPLE wall id 1 xmove 2000.0 save 2000.0 print 1000.0 ps 1000.0

##### 沉积 #####
#停止挤压, 墙的x方向速度改为0.0
WALL id 1 xv 0.0
#沉积。在挤压前端12000~40000.0上方, 沉积约 1 km 颗粒。y的范围需要设置为4000-6000。
#经验: 颗粒充满2km范围, 沉积之后的地层厚度约为1km
GEN NUM 100000.0 rad discrete 60.0 80.0, x ( 10000.0, 41000.0), y ( 4000.0, 6000.0), COLOR gb
↔GROUP sed
#设置沉积颗粒 GROUP=sed 的微观参数
PROP DENSITY 2.5e3, fric 0.3, shear 2.9e9, poiss 0.2, damp 0.4, hertz range GROUP sed
#每 100 步输出一次计算结果
SET print 100
#计算2000步, 让颗粒沉积下来
CYC 2000
#####

#设置墙的挤压速度 x方向速度为2.0
WALL id 1 xv 2.0
#设置墙的挤压量x方向推进3000.0, 每挤压2000.0保存一次计算结果
IMPLE wall id 1 xmove 5000.0 save 5000.0 print 1000.0 ps 1000.0

#计算停止
STOP

```

6.4 绘制最大主应力矢量

接上节有沉积剥蚀实例 `ex_strain_stress`, 完成步骤 1、2、3 和 4, 之后, `zdem` 将生成 `./datass/ss/data/*.out`, 供 GMT 绘图用。目录结构如下

```

|-- ex_strain_stress
|   |-- push_add_del.py
|   |-- data
|       |-- all_0000000000_ini.dat
|       |-- all_0000000000_ini.jpg
|       |-- ...
|       |-- all_0000005000.dat
|       |-- all_0000005000.jpg
|       |-- all_0000058000_ini.dat
|       |-- all_0000058000_ini.jpg
|       |-- all_0000108000.dat
|       |-- all_0000108000.jpg
|-- datass

```

(续下页)

(接上页)

```

|--ss
  |-- data
    |-- *.out
  |-- all_0000006000_ini.dat
  |-- all_0000026000.dat
  |-- all_0000036000_ini.dat
  |-- all_0000056000.dat
  |-- all_0000058000_ini.dat
  |-- all_0000078000.dat
  |-- all_0000108000.dat

```

第 5 步中可以绘制最大主应力矢量。

```

5.zdemss --dir ./datass --strainvol OFF --strainshear ON --sigma1 ON
--scale 1.0 --gridsize 600.0 --stressmean OFF --stressshear OFF
--addball ON --delball ON --xmax 40.0 --ymax 10.0

```

使用 GMT 绘制最大主应力矢量，叠加在变形应变 [strainshear] 上。zdemss 将会读取 zdem 生成的应力应变数据 ./datass/ss/data，生成应力应变云图到 ./datass/ss/*.jpg。设置 x 轴最大值 40.0 km，设置 y 轴最大值 10.0 km。

- --sigma1 ON 绘制最大主应力矢量，注意 --scale 1.0 --gridsize 600.0 是 --sigma1 的子参数，如打算修改 scale 或者 gridsize 的值，必须跟在 --sigma1 后，不可随意调换参数位置。
- --scale 1.0 最大主应力矢量缩放系数，该值越大，矢量越长，大小用户可以根据出图效果自主调节。默认 1.0
- --gridsize 600.0 计算最大主应力均值的正方形边长大小，该值越大，矢量稀疏，大小用户可以根据出图效果自主调节。默认 600.0，可以尝试修改为 1000.0 看看效果。

备注： 参数详解见 [zdemss](#)。如果 scale gridsize 等参数均采用默认值，则可用采用以下命令绘制最大主应力矢量：zdemss --dir ./datass --sigma1 ON --addball ON --delball ON

运行完成之后，目录结构：

```

|-- ex_strain_stress
  |-- push_add_del.py
  |-- data
    |-- all_0000000000_ini.dat
    |-- ...
    |-- all_0000005000.dat
    |-- all_0000058000_ini.dat
    |-- all_0000108000.dat
  |-- datass
    |-- ss

```

(续下页)

(接上页)

```

|-- data
  |-- *.out

|-- data2ps
  |-- strain*.out
  |-- stress*.out

|-- ps
  |-- *.ps

|-- Tmp
  |-- *.grd

  |-- *.jpg
|-- all_0000006000_ini.dat
|-- all_0000026000.dat

```

- ./datass/ss/data/*.out *zdem* 输出的颗粒位置和 (假) 应力的原始数据
- ./datass/ss/data2ps/*.out (真) 应力应变数据, 用于绘制应力应变图 (ps), 用户可以用于定量分析
- ./datass/ss/ps/*.ps 输出的应力应变图 (矢量图)
- ./datass/ss/Tmp/*.grd 计算应力应变产生的中间数据
- ./datass/ss/*.jpg 输出的应力应变图 (位图)

输出的最大主应力矢量见下图:

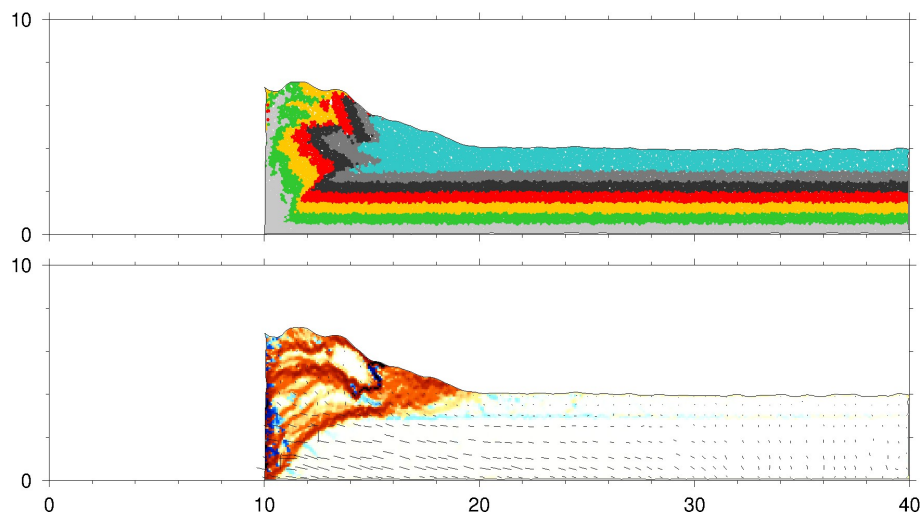


图 6.3: 最大主应力矢量, 叠加在变形应变图上

第 7 章 命令参考

文档约定

所有的命令, 在介绍其用法时, 都尽量遵循如下约定:

- 中括号 [] 括起来的字符串是可选项
- 尖括号 < > 括起来的项表明实际使用时需要用具体的数值替代
- # 之后的内容为注释

比如, `bal [1] id=<int> X=<float> Y=<float> :`

- <int> 是必须的, 使用时需要用具体数字代替
- [1] 是可选的, 实际使用时可以省略

按功能分类

下面将 ZDEM 中的命令按照功能分类, 并用一句话简述其功能。

- 主程序
 - `zdem`: `zdem script.py`, 运行 `script.py` 脚本
- 格式转换、gif 制作
 - `zdem2jpg`: 将计算结果绘制成 jpg 格式。
 - `convert`: 制作 gif
- 绘制应力应变
 - `zdemss`: 绘制应力应变

脚本 `script.py` 中支持的命令如下:

- 程序开始
 - `start`: 开始一个新的计算
 - `restore`: 从某个计算节点恢复, 继续计算
 - `load`: 从坐标文件 `xyr.dat` 文件中读取颗粒坐标和半径, 并生成这些颗粒。
- 颗粒生成
 - `wall`: 基于两个点, 新建一个墙体
 - `ball`: 新建一个颗粒
 - `gen`: 在一个矩形空间中生成一定数量的颗粒
 - `gline`: 在两个点之间建立一组线形排列的颗粒
 - `del`: 删除颗粒, 用于实现剥蚀

- *exp* : 导出颗粒的 *xyr* 等信息到 ASCII 格式文件, 其它离散元软件可以直接读取
- 基本参数设置
 - *set* : 设置计算的基本参数, 如时间步长 *DT* , 重力加速度 *G* 等
- 颗粒参数设置
 - *prop* : 设置颗粒的微观参数 (包括密度、刚度系数、速度等)
 - *init* : 同*prop*
 - *bond* : 断开粘结 *bond*
 - *fix* : 限制颗粒运动
 - *free* : 释放颗粒运动, 与 *fix* 相反
- 范围圈定
 - *range*: 用在 *prop bond del* 等后, 拥有 矩形 *xy* 多边形 *P4* 等圈定方法
- 设置挤压量
 - *cyc* : 设置迭代步数
 - *imple* : 设置挤压距离

ball

说明

`bal [1]` 新建一个颗粒

使用方法

```
BALL ID <int>, X=<float> Y=<float> RAD=<float> [COLOR=<str>] [GROUP=<str>]
```

实例

```
# 生成颗粒, id, 圆心(2.0,0.5), 半径0.5, 蓝色颗粒
BALL ID=0 X=2.0 Y=0.5 RAD=0.5 COLOR=blue
```

bond

说明

`bon [d]` 断开粘结 *bond*

使用方法

```
BOND break RANGE y (<float> <float>)
```

实例

```
#断开y坐标 1~10 的颗粒间粘结
BOND break range y 1.0 10.0
```

convert

说明

读取 jpg tif 等格式的图片, 生成 gif。

可选选项

`-delay=<int>`

播放速度, 一般取 100

实例

```
convert -delay 100 ./data/*[0-9].jpg -loop 0 ./data/process.gif # 制作 gif
```

读取目录 ./data 中的图片 *[0-9].jpg , 并生成 process.gif , 保存到 ./data 中。

cyc

说明

cyc 设置计算步数

使用方法

```
CYC <int>
```

实例

```
#计算5000步  
CYC 5000
```

del

说明

del 删除颗粒, 实现同构造剥蚀。

使用方法

```
DEL RANGE Y (<float> <float>)
```

实例

```
#删除y坐标在 6000.0~99999.0 的颗粒  
DEL RANGE y 6000.0 99999.0
```

exp

说明

exp[xyr] 导出颗粒的 xyr 等信息到 ASCII 格式文件, 其它离散元软件可以直接读取

使用方法

```
#从 xyr.dat 中读取颗粒坐标、半径和 [所属组]  
exp xyr.dat [GROUP] [RANGE ...]
```

其中, xyr.dat 是 x y r [GROUP] 格式的 ASCII 文件, 可由 ZDEM 或者其他软件生成。

实例

```
# 导出全部颗粒的 x y r 信息  
exp xyr.dat  
# 导出全部颗粒的 x y r group 信息  
exp xyr.dat GROUP  
# 导出 bom 组中颗粒的 x y r 信息  
exp xyr.dat range group bom  
# 导出 1000.0 < y < 2000.0 的颗粒的 x y r group 信息  
exp xyr.dat GROUP range y 1000.0 2000.0
```

fix

说明

fix 限制颗粒运动

使用方法

```
FIX x y spin [RANGE ...]
```

实例

```
# 限制颗粒 x y 方向速度和角速度  
FIX x y spin  
# 限制 bom 组中颗粒的 y 向速度和角速度  
FIX y spin RANGE group bom
```

free

说明

free 与 fix 相反, 释放颗粒。

使用方法

```
FREE x y spin [RANGE ...]
```

实例

```
# 释放颗粒 x y 方向速度和角速度
FREE x y spin
# 释放 bom 组中颗粒的 y 向速度和角速度
FREE y spin RANGE group bom
```

gen

说明

gen 在一个矩形空间中生成一定数量的颗粒

- 支持变量 {left} {right} {bottom} {top}, 大括号引用所需要的变量, 如式子 {top}-1000.0, 式子内不能有空格。

使用方法

```
GEN NUM <int>, RAD DISCRETE <float> <float>, X (<float> <float>), Y (<float> <float>) [COLOR=<str>]
↔ [GROUP=<str>]
```

实例

1. 生成一定数量的颗粒, 数量 20000, 颗粒半径随机生成成为 60.0 或者 80.0, 矩形左右边界为 (1000.0, 61000.0), 上下边界为 (1000.0, 13000.0), 黑色

```
GEN NUM 20000 rad discrete 60.0 80.0, x (1000.0, 61000.0), y (1000.0, 13000.0), COLOR black GROUP
↔ball_rand
```

2. 沉积。在挤压顶部往下 1000 米范围内生成颗粒, 沉积后约 500 m 颗粒。经验: 颗粒充满 1km 范围, 沉积之后的地层厚度约为 500m。

```
GEN NUM 100000.0 rad discrete 60.0 80.0, x ({left}, {right}), y ( {top}-1000.0, {top}), COLOR red
↔GROUP sed1
```

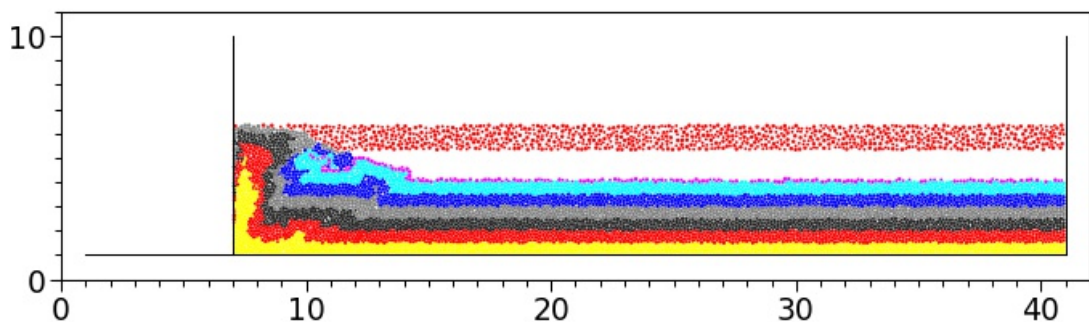


图 7.1: 沉积前

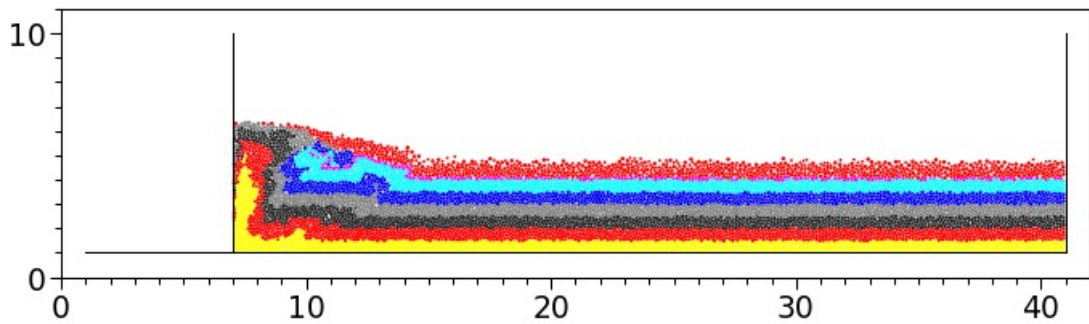


图 7.2: 沉积后

gline

说明

`gline[ne]` 在两个点之间建立一组线形排列的颗粒

使用方法

```
GLINE RAD=<float> P1 ( <float> <float> ) P2 ( <float> <float> ), KN=<float> KS=<float> FRIC=<float>,
↪ [COLOR=<str>] , [GROUP=<str>], [CRATIO=<float>]
```

其中, CRATIO(即 c_{ratio}) 为初始叠和率, $c_{ratio} = |OA|/(r_O + r_A)$, c_{ratio} 取值范围 $(0.0 \ 1.0]$ 。

- 当颗粒刚好接触时, $c_{ratio} = (r_O + r_A)/(r_O + r_A) = 1.0$;
- 当颗粒完全重合时, $c_{ratio} = 0.0/(r_O + r_A) = 0.0$

实例

```
# 在点 ``( 2.0 10.0 )`` 和点 ``( 40.0 10.0 )`` 间生成半径为0.5的颗粒, 法向刚度系数kn=4.
↪14e3, 切向刚度系数ks=4.14e3, 摩擦系数fric=0.55, 颜色color=red, 组名GROUP=bom
GLINE RAD=0.5 P1 ( 2.0 10.0 ) P2 ( 40.0 10.0 ), kn=4.14e3 ks=4.14e3 fric=0.55 color=red GROUP=bom
# 初始叠合率设置为0.8, 颗粒圆心间的距离为|OA| = (r0+rA)*cratio=(0.5+0.5)*0.8=0.8
GLINE RAD=0.5 P1 ( 2.0 10.0 ) P2 ( 40.0 10.0 ), kn=4.14e3 ks=4.14e3 fric=0.55 color=red GROUP=bom
↪CRATIO=0.8
```

imple

说明

`imple[le]` 设置挤压距离

使用方法

```
IMPLE WALL ID <int> xmove <float> print <float> ps <float> save <float>
```

实例

```
#让 id=1的墙，沿着x正方向推进16000.0，每挤压2000.0保存一次计算结果
IMPLE wall id 1 xmove 16000.0 print 2000.0 ps 2000.0 save 2000.0
```

init

说明

ini[t] 同`prop`

load

说明

loa[d] 从 xyr.dat 文件中读取颗粒坐标和半径，并生成这些颗粒。

使用方法

```
#从 xyr.dat 中读取颗粒坐标和半径
LOAD xyr.dat [scale <float>] [x <float> <float> ] [y <float> <float> ]
```

其中, xyr.dat 是 x y r [GROUP] 格式的 ASCII 文件, 可由 ZDEM 或者其他软件生成。

实例

```
# 导入全部颗粒的 x y r [group] 信息，其中 xyr.dat 文件可以包含 GROUP 信息
LOAD xyr.dat
# 导入颗粒的 x y r 信息，并缩放 0.1 倍，即 x=scale*x, y=scale*y, r=scale*r
LOAD xyr.dat scale 0.1
# 导入颗粒 (1000.0<x<2000.0 并且 2000.0<y<4000.0) 的 x y r 信息
LOAD xyr.dat x 1000.0 2000.0 y 2000.0 4000.0
```

prop

说明

pro[p] 设置材料参数

可选选项

DENSITY=<float>

颗粒密度

FRIC =<float>

颗粒密度

SHEAR=<float>

颗粒密度

POISS=<float>

颗粒密度

DAMP=<float>

局部阻尼系数, 见 PFC2D 4.0 手册或李长圣博士论文

HERTZ

使用 hertz 接触模型

xv=<float>

颗粒 x 方向速度

yv=<float>

颗粒 y 方向速度

spin=<float>

颗粒角速度

ebmod=<float>

粘结的杨氏模量, ref. [李长圣博士论文,2019]

gbmod=<float>

粘结的剪切模量, ref. [李长圣博士论文,2019]

tstrength=<float>

粘结的抗拉强度, ref. [李长圣博士论文,2019]

sstrength=<float>

粘结的聚合强度, ref. [李长圣博士论文,2019]

n_bond=<float>

粘结的抗拉强度, ref. [PFC2D 4.0 手册]

s_bond=<float>

粘结的抗剪强度, ref. [PFC2D 4.0 手册]

使用方法

```
PROP DENSITY <float>, FRIC <float>, SHEAR <float>, POISS <float>, DAMP <float>, HERTZ [RANGE ...]
PROP ebmod <float>, gbmod <float>, tstrength <float>, sstrength <float>, [tolerate 10.0] [fric
↔<float>] [RANGE ...]
PROP DENSITY <float>, FRIC <float>, KN <float>, KS <float> [tolerate rext 0.8] [RANGE ...]
PROP n_bond <float> s_bond <float> [RANGE ...]
PROP xv <float> yv <float> spin <float> [RANGE ...]
```

其中, `tolerate` 默认值为 $1e-6$, `rext` 默认值为 1.0 。 `rext` 可与 `gline` 的 `cratio`, 联合使用, 设置 `rext=cratio`, 可以避免 0-1-2-3-4... 线性叠合的颗粒, 出现 0-2 跨颗粒 1 产生粘结。

实例

```
# Model 1. ref. [Morgan 2015 JGR] 和 [李长圣博士论文,2019]
# 设置材料参数 密度DEM, 摩擦系数FRIC, 剪切模量shear, 泊松比poiss, 阻尼DAMP
PROP DENSITY 2.5e3, fric 0.3, shear 2.9e9, poiss 0.2, damp 0.4, HERTZ
#设置 range x 1.0 10.0 y 1.0 10.0 中颗粒粘结在一起, 使颗粒具有粘聚力
PROP ebmod 2e8 gbmod 2e8 tstrength 2e7 sstrength 4e7 fric 0.3 range x 1.0 10.0 y 1.0 10.0
#
# Model 2. ref. [PFC2D 4.0 手册] 和 [李长圣博士论文,2019]
# 设置所有颗粒的材料参数 密度DEM, 摩擦系数FRIC, 法向刚度KN, 切向刚度KS, 阻尼DAMP
PROP DENSITY 2.5e3, KN 1e4, KS 1e4, FRIC 0.3, DAMP 0.4
```

(续下页)

(接上页)

```
#
#
#设置 (|OA| - (rO+rA)|) <= tolerate的颗粒粘结在一起, 使颗粒具有粘聚力, 使得更多颗粒粘结到一起
PROP n_bond 1.0e4 s_bond 1.0e4 tolerate 10.0
#
#设置 (|OA| - (rO+rA)*rest|) <= tolerate的颗粒粘结在一起, 使颗粒具有粘聚力
↔。这里, tolerate取默认值1e-6
PROP n_bond 1.0e4 s_bond 1.0e4 tolerate rest 0.8

#设置所有颗粒线速度和角速度为零
PROP xv 0.0 yv 0.0 spin 0.0
```

range

简介

圈定颗粒范围

不指定 range 则默认选定全部颗粒, 必须放在其他命令的最后 prop bond del 后。

圈定方法

- 矩形 X (<float> <float>), Y (<float> <float>), 指定 左右边界 下上边界
- 四边形 P4 (<float> <float>), (<float> <float>), (<float> <float>), (<float> <float>), 逆时针指定四个点
- 椭圆 ellipse (<float> <float>), <float> <float>, 指定椭圆中心 (x0, y0) 和 a、b, 椭圆方程 $(x-x_0)^2/a^2 + (y-y_0)^2/b^2 = 1$
- 组名 GROUP <str>
- 支持变量 {left} {right} {bottom} {top}, 大括号引用所需要的变量, 如式子 {top}-1000.0, 式子内不能有空格。

实例

1. 矩形

```
#圈定一个矩形范围, 左右边界(1.0 10.0), 下上边界(1.0 10.0)
#将该范围内的颗粒的颜色设置为红色
PROP COLOR red RANGE x (1.0 10.0) y (1.0 10.0)
```

2. 四边形

```
#圈定一个四边形范围, 逆时针指定四个点 ( 2e-2, 0.0) (3e-2, 0.0) ( 1e-2, 2e-2) (1e-2, 1e-2)
#将该范围内的颗粒的摩擦系数设置为0.0
PROP fric 0.0 range P4 ( 2e-2, 0.0) (3e-2, 0.0) ( 1e-2, 2e-2) (1e-2, 1e-2)
```

3. 椭圆

```
#固定一个椭圆范围，指定椭圆中心和a,b ( 2e-2, 0.0) 3e-2, 0.0
#将该范围内的颗粒的摩擦系数设置为0.0
PROP fric 0.0 range ellipse ( 2e-2, 0.0) 3e-2, 0.0
```

4. 使用变量 {left} {right} {bottom} {top}

```
#删除顶部1km的颗粒
DEL RANGE y {top}-1000.0, {top}
```

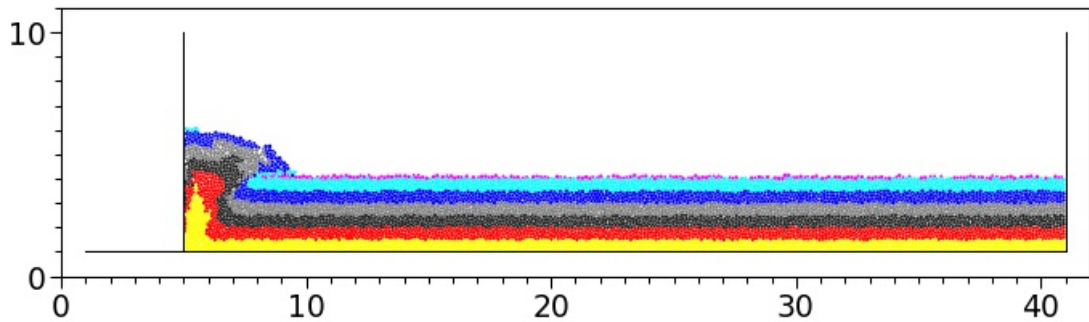


图 7.3: 剥蚀前

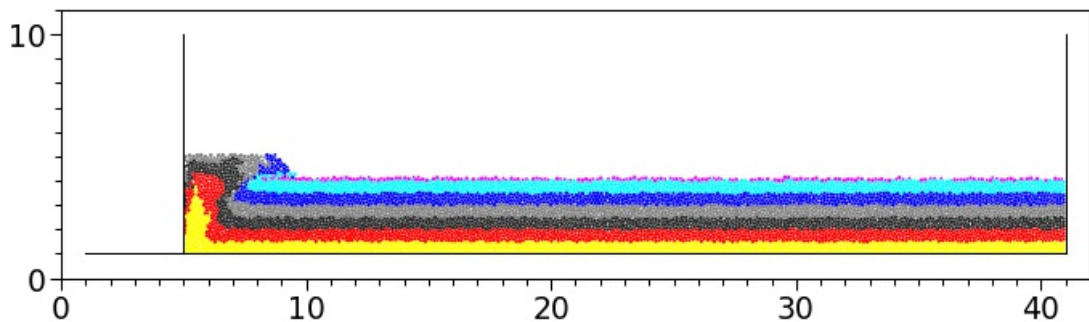


图 7.4: 剥蚀后

使用 AND OR

- AND 并

```
#颗粒在矩形X()Y()并且在GROUP中，才设置为红色
prop color red range X (<float> <float>), Y (<float> <float>) AND GROUP <str>
```

- OR 或

```
#颗粒在矩形X()Y()或者在GROUP中，都设置为红色
prop coloe red range X (<float> <float>), Y (<float> <float>) OR GROUP <str>
```

实例

1. AND

```
# 颗粒在矩形  $x()$   $y()$ 中, 并且在四边形  $P4 ()()()$ 中, 则其颜色设置为红色
PROP COLOR red RANGE x (1.0 10.0) y (1.0 10.0) AND P4 ( 2e-2, 0.0) (3e-2, 0.0) ( 1e-2, 2e-2) (1e-2,
↪1e-2)
#颗粒在矩形  $x()$   $y()$ 中, 并且在组 GROUP sed中, 则其颜色设置为红色
PROP COLOR red RANGE x (1.0 10.0) y (1.0 10.0) AND GROUP sed
```

2. OR

```
#颗粒在矩形  $x()$   $y()$ 中, 或者在  $P4 ()()()$ 中, 其颜色设置为红色
PROP COLOR red RANGE x ( 1e-2, 2.0e-2) y ( 0e-2, 1.5e-2) OR P4 ( 2e-2, 0.0) (3e-2, 0.0) ( 1e-2, 2e-
↪2) (1e-2, 1e-2)
```

restore

说明

res[tore] 从某个计算节点恢复, 继续计算。

使用方法

```
#从 ``00200.sav`` 中恢复计算
RES 00200.sav
```

其中, 00200.sav 是 ZDEM 保存的一个计算节点。

set

说明

设置计算的基本参数, 如时间步长 DT, 重力加速度 G 等

使用方法

设置时间步长和重力加速度:

```
SET DT <float>, GRAVITY (<float> <float>)
```

设置颗粒形状, 默认为 on, 即圆盘颗粒体积为 $V = \pi \cdot r^2$; off 为球, 颗粒体积为 $V = 4/3 \cdot \pi \cdot r^3$

```
SET disk <on|off>
```

设置进度条刷新闻隔:

```
SET stepbar <int>
```

设置 ZDEM 格式的 .sav 文件保存间隔:

```
SET sav <int>
```

设置 .ps 矢量图保存间隔:

```
SET ps <int>
```

设置 ZDEM 格式的 .dat ASCII 文件保存间隔:

```
SET print <int>
```

设置 paraview 格式的 .vtk 文件保存间隔:

```
SET vtk <int>
```

实例

```
#设置时间步DT 及 重力加速度 G
SET DT 5e-2, GRAVITY 0.0, -9.8
#设置颗粒形状
SET disk off #球, 计算颗粒体积用 $4/3*pi*r^3$ 计算
SET disk on #圆盘 (默认开启),  $V=pi*r^2$ 
#每计算 100 步更新一次进度条
SET stepbar 100
```

start

说明

star[rt] 开始一个新的计算, 申请内存、初始化变量。

使用方法

```
START
```

wall

说明

wal[l] 基于两个点, 新建一个墙体

使用方法:

```
WALL ID <int>, NODES ( <float> <float> ) ( <float> <float> ), KN=<float> KS=<float> FRIC=<float>,  $\mu$ 
↔ [COLOR=<str>]
WALL ID <int>, XV <float>, YV <float>
```

注意: 墙为一个射线, 射线仅左边受力。

实例:

```
# 建立一个墙, id为1, 两个点( 2.0 40.0 ) ( 2.0 2.0 )连线确定该墙, 法向刚度系数kn=4.
↔14e3, 切向刚度系数ks=4.14e3, 摩擦系数fric=0.55, 颜色color=red
WALL id 1, nodes ( 2.0 40.0 ) ( 2.0 2.0 ), kn=4.14e3 ks=4.14e3 fric=0.55 color=red
# 设置墙的挤压速度 x方向速度为2.0
WALL id 1 xv 2.0
# 设置墙的挤压速度 x方向速度为0.0 y方向速度为2.0
WALL id 1 xv 0.0 yv 2.0
```

zdem

说明

zdem 主程序。

使用方法

zdem [可选项] [script.py]

script.py 为 ASCII 码格式文件, 例如 `zdem script.py`, 即可运行脚本 script.py。

可选项

--addball

配合-s 选项, 应力应变计算过程中, 有新颗粒加入体系 (沉积), 默认关闭。

--delball

配合-s 选项, 应力应变计算过程中, 删除了颗粒 (剥蚀), 默认关闭。

-g, --grid SIZE

配合-s 选项, 设置应力应变计算时候, 网格的大小 SIZE, 默认 200.0

-h, --hlep

显示帮助信息

-j NUM

并行计算开辟的线程数 NUM, 默认使用 OMP_NUM_THREADS, 如未设置 OMP_NUM_THREADS, 则使用全部 CPU 物理核心数并行计算。

--leftwallid ID

配合-s 选项, 设置左边墙 ID, 该墙左边颗粒均会被删除。如果颗粒被挤出到左边墙之外, 需要设置该参数。

--rightwallid ID

配合-s 选项, 设置右边墙 ID, 该墙右边颗粒均会被删除。如果颗粒被挤出到右边墙之外, 需要设置该参数。

-s, --strin-stres DataDir

计算应力应变从 DataDir 读取数据, 将应力应变输出到 DataDir/ss 目录

-v, --version

打印版本信息

--xmove X

配合-s 选项, 设置模型 x 方向偏移位移 X, 默认 0.0。

`--ymove Y`

配合-s 选项, 设置模型 y 方向偏移位移 Y, 默认 0.0。

实例

`zdem -h`

打印帮助文档

`zdem -j 8 script.py`

使用 8 个线程并行计算 `script.py`

`zdem -s ./data`

读取 `./data` 中的数据计算应力应变, 应力应变数据默认保存到了 `./data/ss/data` 中。参考应力应变[无沉积剥蚀](#) 和[有沉积剥蚀](#)

`zdem --xmove -1000.0 --ymove -1000.0 -s ./data`

将模型向左移动 1000.0, 向下移动 1000.0, 之后, 计算应力应变。

zdemss

说明

绘制应力应变。

使用方法

`zdemss` [可选项]

例如 `zdemss --dir ./datass` , 将 `./datass` 目录中的数据绘制成云图。前提, 已经采用 `zdem -s ./datass` 计算了应力应变数据, 数据已默认保存到了 `./datass/ss/data` 中, 见[zdem](#)。zdemss 完成应力应变绘制, 生成的目录结构如下:

- `./datass/ss/data` 该文件夹保存了 `zdem` 输出的颗粒位置和 (假) 应力的原始数据
- `./datass/ss/data2ps` 该文件夹内为 (真) 应力应变数据, 用于绘制应力应变图 (ps), 用户可以用于定量分析
- `./datass/ss/ps` 该文件夹内为应力应变图 (矢量图)
- `./datass/ss/Tmp` 该文件夹内为计算应力应变产生的中间数据
- `./datass/ss/*.jpg` 绘制完成的应力应变图 (位图)

可选项

`--addball ON/OFF`

有沉积事件, 默认 OFF

`--delball ON/OFF`

有剥蚀事件, 默认 OFF

`-d, --dir`

设置数据所在目录

`-h, --help`

打印帮助信息

`--showcolorbar ON/OFF`

绘制颜色条, 默认 ON

--showlable ON/OFF/abc
绘制颜色条, 默认 OFF, 其中 abc 只给子图命名为 abc

--strainvol ON/OFF
绘制体积应变, 默认 ON

--strainshear ON/OFF
绘制变形应变, 默认 ON

--straincolormap value
设置应变颜色映射表, 默认 default。 示例 **--straincolormap ./ss/gradwhite-invar.cpt**, 其中 **./ss/gradwhite-invar.cpt** 为用户自定义的颜色映射表

--sigma1 ON/OFF
最大主应力矢量 (叠加在应变图上), 默认 OFF 不绘制

- **--scale value**
矢量长度缩放系数, 默认 1.0(取值范围 0.0~+∞)
- **--gridsize value**
计算最大主应力均值的正方形边长大小, 默认 600.0(取值应大于颗粒直径)。取值越大, 绘制的矢量线越稀疏。

--stressmean ON/OFF
绘制平均应力, 默认 ON

--stressshear ON/OFF
绘制剪切应力, 默认 ON

--width value
图片宽 (cm), 默认 14

--xmax value
x 轴最大值 (km), 默认自动设置

--ymax value
y 轴最大值 (km), 默认自动设置

--maxstress value
最大应力值 (MPa), 默认 300

-v, --version
显示版本信息

实例

zdemss --dir ./data

读取 ./data 中的数据计算应力应变。

zdemss --dir ./datass --xmax 40.0 --ymax 10.0 --maxstress 250.0

读取 ./datass 中的数据计算应力应变, 设置 x 轴最大值 40.0 km, 设置 y 轴最大值 10.0 km, 设置颜色条应力最大值 400 MPa

zdemss --dir ./data --sigma1 ON --scale 2.0 --gridsize 1000.0

读取 ./data 中的数据计算应力应变, 绘制最大主应力矢量, 默认叠加在变形应变图

上

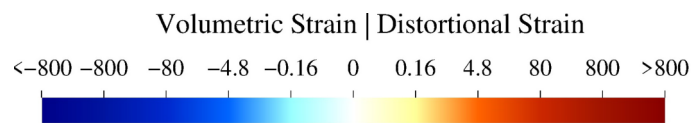
备注：详细用法示例，参考应力应变[无沉积剥蚀](#)、[有沉积剥蚀](#)和[绘制最大主应力矢量](#)

高级功能

`--straincolormap default/filename.cpt`

绘制颜色映射表，默认 default，示例 `--straincolormap ./data/ss/gradwhite-invar.cpt`

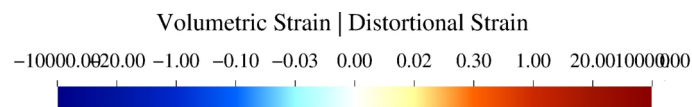
- 示例 1



`./data/ss/gradwhite-invar.cpt` 内容如下：

-16000	0	0	136	-800	0	20	160
-800	0	20	160	-80	0	40	204
-80	0	40	204	-4.8	0	100	255
-4.8	0	100	255	-0.16	153	255	255
-0.16	153	255	255	0			
↔255	255	255					
0	255	255	255	0.			
↔16	255	255	153				
0.16	255	255	153	4.8	255	100	0
4.8	255	100	0	80	204	40	0
↔							
80	204	40	0	800	160	20	0
800	160	20	0	16000	136	0	0
B	0	0	0				
F	0	0	0				
N	128	128	128				

- 示例 2



`./data/ss/gradwhite-invar2.cpt` 中内容如下：

-10000	0	0	136	-20.			
↔0	0	20	160				
-20.0	0	20	160	-1.0	0	40	204
-1.0	0	40	204	-0.1	0	100	255
-0.1	0	100	255	-0.03	153	255	255

(续下页)

(接上页)

-0.03	153	255	255	0	255	255	255
0	255	255	255	0.02	255	255	153
0.02	255	255	153	0.3	255	100	0
0.3	255	100	0	1.0	204	40	0
1.0	204	40	0	20.0	160	20	0
20.0	160	20	0	10000	136	0	0
B	0	0	0				
F	0	0	0				
N	128	128	128				

备注:

- 颜色映射表内容解释参考 <https://docs.gmt-china.org/5.4/cpt/format/>
- 颜色映射表制作方法参考 <https://docs.gmt-china.org/5.4/cpt/makecpt/>

zdem2jpg**说明**

读取软件生成的 all_*.dat 文件绘制如 jpg 格式的图片

实例

```
zdem2jpg --dir=./data --ymin=0.0 --xmin=0.0 --xmove=-1000.0 --ymove=-1000.0 --linewidth=0.8 --
↔fontsize=8 --xmax=50000.0 --ymax=10000.0 --dpi=600 --pagesize=14 --topshow=false --rightshow=false
```

必选选项**--dir=<dir>**

指定 all_*.dat 数据所在目录, <dir> 为 all_*.dat 所在目录。例如 “./data”

可选选项**--xmax=<float>**

设置绘图坐标 x 轴最大值

--ymax=<float>

设置绘图坐标 Y 轴最大值

--xmin=<float>

绘图 X 最小值, 默认 0.0

--ymin=<float>

绘图 Y 最小值, 默认 0.0

--xmove=<float>

坐标沿 x 轴偏移量, 默认 0.0

--ymove=<float>

坐标沿 y 轴偏移量, 默认 0.0

--major_locator=<float>
主坐标刻度间隔, 默认 10000.0

--minor_locator=<float>
次坐标刻度间隔, 默认 1000.0

--fontsize=<int>
坐标刻度字体大小, 默认 9

--max_workers=<float>
并行进程数, 默认 24

--dpi=<int>
图片分辨率, 默认 600

--linewidth=<float>
线条粗细, 默认 0.8

--pagesize=<int>
图片大小, 单位 cm, 默认 14

--leftshow=<bool>
显示坐标轴左线框, 取值 true/false, 默认 true

--rightshow=<bool>
显示坐标轴右线框, 取值 true/false, 默认 true

--bottomshow=<bool>
显示坐标轴底线框, 取值 true/false, 默认 true

--topshow=<bool>
显示坐标轴顶线框, 取值 true/false, 默认 true

--wallshow=<bool>
显示 wall 墙, 取值 true/false, 默认 true

--surfaceshow=<bool>
显示变形边界, 取值 true/false, 默认 true

--bondplot=<bool>
显示粘结连接关系, 取值 true/false, 默认 false, 黑色是没有粘结状态, 红色是有粘结拉伸状态, 蓝色是有粘结挤压状态

--colormap=<str>
指定颜色配置文件, 格式为 10x3 的矩阵, 对应十个 RGB 值, 默认取值及文件格式见 [颜色表](#)。建议直接制定该文件的绝对路径或者相对路径, 如 `--colormap=/home/zhangsang/MyColorMap.txt` 或 `--colormap=./MyColorMap.txt`。如果仅指定文件名, 如 `--colormap= MyColorMap.txt`, 搜索顺序为当前目录 > `-dir` 指定的目录 > Home 目录。

• **实例** `zdem2jpg --dir=./data`

读取目录 `./data` 中的计算数据 `all_*.dat`, 并生成 jpg 格式的图片, 保存到 `./data` 中。

• **实例** `zdem2jpg --dir=./data --xmax=40000.0 --ymax=10000.0`

读取目录 `./data` 中的计算数据 `all_*.dat`, 并生成 jpg 格式的图片, 保存到 `./data` 中。图片 X 轴最大值为 40000.0, y 轴最大值为 10000.0

第 8 章 离散元原理

二十世纪七十年代, 基于分子动力学原理, [Cundall1979] 首先提出了对颗粒物质进行离散模拟的思路, 用于研究岩土体的各种力学行为, 其基本思想是将颗粒材料内部细观尺度的单个离散颗粒视为一个离散单元, 将颗粒集合体模型视为若干离散单元的集合, 通过一系列离散的单元来模拟颗粒材料的力学行为。离散元法假设单元为刚性的, 不可变形; 单元相互作用通过弹簧阻尼器和滑动摩擦器产生的力体现; 每一个时步内, 速度和加速度是常数, 这要求时间步足够小, 同时保证一个时步内, 单元只能以很小的位移与其相邻单元相互作用。

8.1 最小的离散元程序

[Zhao2015] 用 45 行 MATLAB 的离散元代码模拟了伽利略的比萨斜塔试验, 其前处理、求解器和后处理代码 [Code 7.1](#) :

代码 7.1 伽利略的比萨斜塔试验 (MATLAB)

```
% function HelloDEM % Galileo' s leaning Tower of Pisa experiment
M_Ball_Steel=10.0; % 10 kg steel ball
M_Ball_Wood=1.0; % 1 kg wood ball
Y0_Ball_Steel = 100.0; % initial position
Y0_Ball_Wood = 100.0;
dT = 0.01; % time step
Nloops = 500; % number of cycles
G = 9.8; % gravitational constant
U_Ball_Steel = 0.0; % initial displacement
U_Ball_Wood = 0.0;
V_Ball_Steel = 0.0; % initial velocity
V_Ball_Wood = 0.0;
A_Ball_Steel = 0.0; % accelerated velocity
A_Ball_Wood = 0.0;
Y_Ball_Steel = Y0_Ball_Steel; % initial position
Y_Ball_Wood = Y0_Ball_Wood;
t = 0.0;
t_H = []; % history of time
Y_H_Ball_Steel = [];
Y_H_Ball_Wood = [];
for i=1:Nloops
    F_Ball_Steel=-G*M_Ball_Steel; %ball force, constitutive model
    F_Ball_Wood = -G*M_Ball_Wood;
    A_Ball_Steel= F_Ball_Steel/M_Ball_Steel; %Newton' s second law
    A_Ball_Wood = F_Ball_Wood /M_Ball_Wood;
    V_Ball_Steel= V_Ball_Steel+A_Ball_Steel*dT; %update velocity
    V_Ball_Wood = V_Ball_Wood+A_Ball_Wood*dT;
```

(续下页)

(接上页)

```

U_Ball_Steel=U_Ball_Steel+V_Ball_Steel*dT; %update displacement
U_Ball_Wood = U_Ball_Wood+ V_Ball_Wood*dT;
Y_Ball_Steel = YO_Ball_Steel + U_Ball_Steel; %update position
Y_Ball_Wood = YO_Ball_Wood + U_Ball_Wood;
t=t+dT;
%record history
t_H=[t_H,t];
Y_H_Ball_Steel=[Y_H_Ball_Steel,Y_Ball_Steel];
Y_H_Ball_Wood=[Y_H_Ball_Wood,Y_Ball_Wood];
end
plot(t_H,Y_H_Ball_Steel,'k','linewidth',1);
hold on;
plot(t_H(1:20:end),Y_H_Ball_Wood(1:20:end),'ks','MarkerFace','y');
Ground=zeros(1,length(t_H));
plot(t_H,Ground,'r');
xlabel('Time [s]')
ylabel('Height [m]')
le=legend('Steel Ball','Wood Ball','Ground');

```

比萨斜塔离散元模拟试验中，钢球和木球高度随时间的变化结果见 *Fig 7.1*，可见钢球和木球是同时落地的。

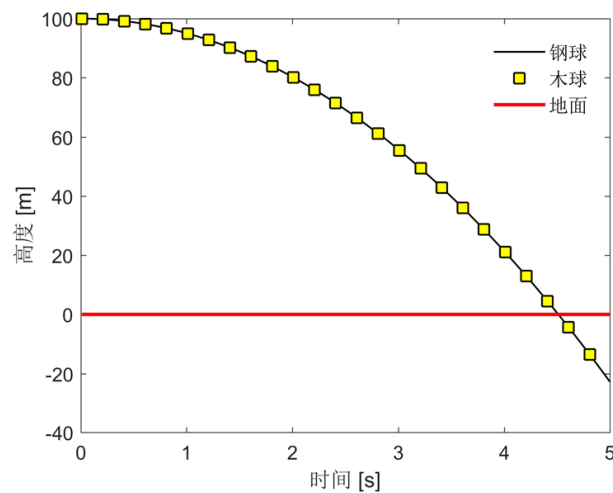


图 8.1: 球高度随时间的变化，比萨斜塔离散元模拟试验。



图 8.2: 球高度随时间的变化，比萨斜塔离散元模拟试验。

离散元的求解实际上是迭代计算颗粒位移和受力，可以概括为两部分，如 *Fig 7.2* 所示。

第一步, 已知的颗粒所受合力和合力矩, 由牛顿第二定律更新每个颗粒的位置 (如代码 [Code 7.1](#) 中, 24~31 行); 第二步, 找到相互接触的颗粒, 应用接触力学模型 (即力-位移法则, 或者称本构模型。如代码 [Code 7.1](#) 中, $F_{\text{Ball_Steel}} = -G \cdot M_{\text{Ball_Steel}}$, 该实例中, 没有颗粒相互作用, 颗粒仅受重力作用, 实际这里需要计算颗粒间的相互作用力) 计算颗粒间的接触力, 进而得到所有颗粒受到的合力与合力矩。反复执行这两个步骤, 直到计算结束。

8.2 颗粒位置的更新

得到某颗粒在时刻 t 所受合力后, 就可以更新颗粒的位置和角速度。根据牛顿第二运动定律有:

$$\vec{a}_t = \vec{F}_t / m$$

$$\eta_t = M_t / I$$

其中, \vec{a}_t 和 η_t 分别表示颗粒在时刻 t 的平动加速度和角加速度, \vec{F}_t 和 M_t 分别为颗粒在该时刻的合力和合力矩, m 和 I 分别为颗粒的质量和惯性矩。

假设已知某颗粒在 η_t 时的位置 \vec{P}_t 和 $t - \Delta t$ 平动加速度 $\overrightarrow{v_{t-\Delta t/2}}$ 时的平动速度, 则按照跳蛙法有,

$$\overrightarrow{v_{t+\Delta t/2}} = \overrightarrow{v_{t-\Delta t/2}} + \vec{a}_t \cdot \Delta t$$

$$\overrightarrow{P_{t+\Delta t}} = \vec{P}_t + \overrightarrow{v_{t+\Delta t/2}} \cdot \Delta t$$

其中, Δt 为一个时步。

首先, 得到颗粒在 $t + \Delta t/2$ 时的速度 $\overrightarrow{v_{t+\Delta t/2}}$, 再得到颗粒在 $t + \Delta t$ 时的位置 $\overrightarrow{v_{t+\Delta t/2}}$ 。之后, 由相应的接触力学模型计算颗粒间的接触力, 得到 $t + \Delta t$ 时颗粒所受合力, 通过牛顿第二运动定律求得 $t + \Delta t$ 的颗粒的平动加速度, 就可以更新颗粒 $t + \Delta t$ 时的位置, 这样一直迭代循环下去, 直到计算结束。

同理, 假设某颗粒在 t 时的角度为 θ_t 和角加速度为 η_t , $t - \Delta t/2$ 时的角速度为 $\omega_{t-\Delta t/2}$, 则按照跳蛙法有,

$$\omega_{t+\Delta t/2} = \omega_{t-\Delta t/2} + \eta_t \cdot \Delta t$$

$$\theta_{t+\Delta t} = \theta_t + \omega_{t+\Delta t/2} \cdot \Delta t$$

在接触力学模型部分, 颗粒所受切向力一般由位移增量 (颗粒在接触点处的相对速度 $\times \Delta t$) 计算得到, 不需要颗粒的角度值。如果没有其它特殊需要, 可以不更新颗粒的角度, 只更新颗粒的角速度即可。

8.3 接触力的计算

自从 Cundall 和 Strack 给出了对颗粒物质进行离散模拟的思路，各个学科领域的学者提出了各种适用于不同问题的 DEM 接触力学模型。自然界中，颗粒间的作用力往往是非线性的，真实的再现自然界中的颗粒行为，需要构建复杂的接触力学模型，相应的求解接触力也需要更大的计算量。然而，一些简化的接触力学模型，也可以解释一些自然界中的岩土体相互作用的机制，如断层传播时的脉冲滑动行为 [Place1999]、高孔隙度砂岩中不同形态压密带的形成机制 [Liu2015] 等。这里，我们将详细阐述每种接触模型的实现方法，以往研究中，不同的接触模型往往采用不同的粘结模型。将粘结模型和接触模型作为一整体，将从接触模型、粘结模型和计算实例三个方面给出实现方法。

首先区分一下粘结和接触的区别，如果两颗粒间的圆心距离小于等于两颗粒半径之和，则认为两颗粒是相互接触的。离散元初始模型生成后，相互接触的颗粒间生成粘结，粘结的信息放入粘结链表，粘结有抗拉和剪切强度，整个计算过程粘结往往只生成一次。两个颗粒间的粘结断裂之后，如果再次接触，则接触信息放入无粘结接触链表，无粘结接触没有抗拉和抗剪强度 (Fig 7.3)。

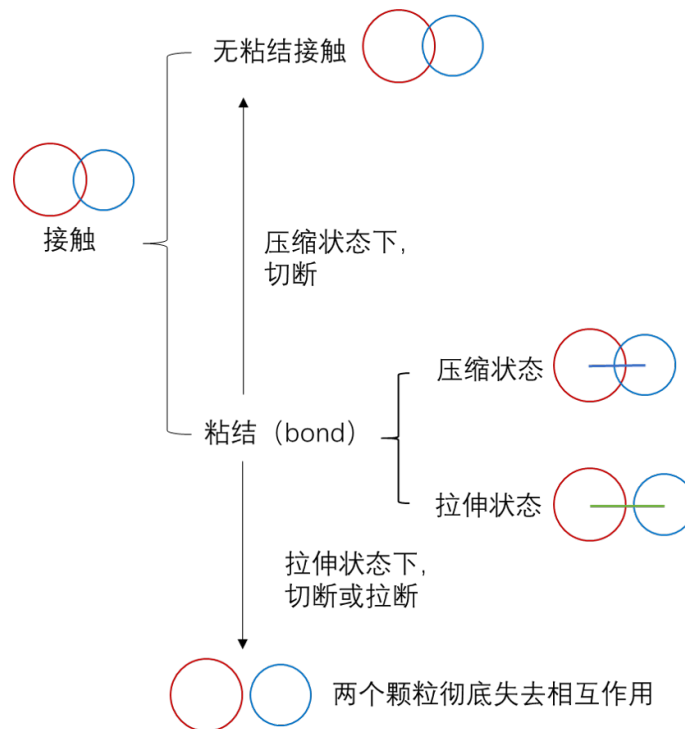


图 8.3: 接触与粘结关系

本文涉及的接触模型分成两类线弹性模型和 Hertz-Mindlin 模型。

8.3.1 线弹性模型

Cundall 和 Strack(Cundall and Strack,1979) 于 1979 年的给出线弹性模型, 该模型是 DEM 中最基本的接触力学模型, 基本上所有的离散元软件中嵌入了该模型 (Cundall and Strack,1979; Cundall P A,1988a; Itasca Consulting Group,2008; Itasca Consulting Group,2012b; Šmilauer et al.,2018)。与之后 PFC2D(Itasca Consulting Group,2008) 中接触点的描述略微不同。这里采用 PFC2D 中的描述的线弹性接触模型。详见 [Li2019]

8.3.2 Hertz-Mindlin 模型

Hertz 模型中, 颗粒为一个球, 而不再是之前的圆盘。模型提供了颗粒挤压状态下的法向力的计算方法, 不提供拉伸状态下法向力计算方法。Cundall 描述的 Hertz_Mindlin 接触模型 (Cundall P A,1988a) 是 Mindlin 和 Deresiewicz 理论模型 (Mindlin,1953) 的近似, 是一种非线性接触模型。法向力采用公式 (2-12) 计算, 切向力采用公式 (2-15) 计算, 考虑颗粒转动。详见 [Li2019]

第 9 章 颜色表

软件支持的颜色列表如下，支持 **数字**和 **字符**两种方式：

数字	字符	颜色	RGB
0	lg	light gray	0.85
1	green	green	010
2	yellow	yellow	110
3	red	red	100
4	white	white	1.0
5	black	black	0.15
6	mg	medium gray	0.5
7	blue	blue	001
8	gb	green/blue	011
9	violet	violet	101

后处理阶段，使用 `zdem2jpg` 命令时，可以通过 `--colormap` 制定自己的 RGB 格式的颜色配置表，默认颜色配置为 `ColorRicebal.txt` 格式如下：

```
0.85 0.85 0.85
0.00 1.00 0.00
1.00 1.00 0.00
1.00 0.00 0.00
0.90 0.90 0.90
0.15 0.15 0.15
0.50 0.50 0.50
0.00 0.00 1.00
0.00 1.00 1.00
1.00 0.00 1.00
```



第 10 章 开通账户

使用软件需要遵守用户协议, 见[使用协议](#)。

申请 [并行超算云](#) 帐号, 请联系 [北京并行科技股份有限公司](#) 的

周凤	客户经理
手机	18627783589
邮箱	zhoufeng@paratera.com

第 11 章 Linux 命令行

使用软件应该具备 Linux 命令行基本操作知识。

- 查看当前所在目录

```
pwd
```

输出 `/share/home/zhangsan`

- 列出当前目录下的目录和文件

```
ls
```

输出 `backup.sh bin Desktop git help programs projects vbox`

- 改变当前目录

到当前目录下的 `Desktop` 目录里

```
cd Desktop
```

到上一级目录。当前目录 `.` 上级目录 `..`

```
cd ..
```

- 创建目录

在当前目录下创建一个 `data` 目录

```
mkdir data
```

- 删除文件或目录

删除当前目录下 `a.txt`

```
rm a.txt
```

递归删除当前目录下的 `data` 目录中的全部内容和 `data` 目录本身

```
rm -r data
```

- 移动或换名 `mv < 源文件或原目录> < 目标文件或目标目录>`

```
mv vi.tex vi.txt #把文件vi.tex 换名为vi.txt
mv data data1 #把目录data 换名为data1
mv data .. #把目录data 移动到上级目录
```

- 复制文件或目录 `cp < 参数> [源文件] [目标文件或路径]`

```
cp a.txt b.txt # 把当前目录下的a.txt复制一份，并命名为b.txt
cp -r data ../maya # 把当前目录下的data目录复制到上一级目录下的maya 目录中
```

- 分屏查看文件内容 `more, less < 参数 > < 文件名 >`

分屏查看文件 `example.txt` 的内容。按空格键向下翻页，b 向上翻页

```
more example.txt
```

- 改变文件属性 `chmod < 参数> < 属性> < 文件名或目录>`

将文件 `example.txt` 改成自己可读、可写、可执行同组和其他用户只准读和执行

```
chmod 755 exaple.txt
```

备注： 更详细的教程请参考 [Linux 命令行基本操作知识](#)。

第 12 章 使用协议

ZDEM 2.1 用户使用协议

2021-06-17 用户使用 ZDEM 2.1, 默认同意本协议。

ZDEM 2.0 是一个用于构造变形研究的二维离散元软件。采用 C 语言编写, 基于 OpenMP 实现并行计算, 主要用于构造变形数值模拟, 补充构造物理沙箱实验在应力应变及材料选取上的局限性, 为构造变形研究提供一种新的方法。

用户 发表论文、作会议报告时, 须注明数值模拟使用了 ZDEM, 并按照以下示例致谢并引用相关文献。

- 致谢示例

模拟实验使用的软件为李长圣博士研发的离散元数值模拟软件ZDEM(<https://geovbox.com>), 应力应变的处理参考了Julia Morgan提供的脚本, 在此表示感谢。

另外, 致谢并行科技股份有限公司可申请奖励机时。

- 中文版

感谢并行科技股份有限公司提供的HPC计算资源与服务, 公司官网: <https://paratera.com>

- 英文版

The authors acknowledge Beijing PARATERA Tech CO.,Ltd. for providing HPC resources that have contributed to the research results reported within this paper. URL: <https://paratera.com/>

- 论文引用

1. 李长圣 (2019) 基于离散元的褶皱冲断带构造变形定量分析与模拟. 博士论文. 南京大学. (必须引用)
2. Li, C.S., Yin, H.W.*, et al. (2021) Calibration of the discrete element method and modelling of shortening experiments. *Front. Earth Sci.* 9:636512. (可选, 线弹性接触模型、应变分析)
3. LI C.S., YIN H.W.*, et al. (2018) Validation Tests for Discrete Element Codes Using Single-Contact Systems. *International Journal of Geomechanics* 18, 06018011.7. (可选, 偏重代码测试)
4. 李长圣, 尹宏伟 *, 等. (2017) 共享内存式并行离散元程序的设计与测试. *南京大学学报 (自然科学)*, (06): 1161-1170. (可选, 偏重软件开发)

-
5. Morgan J.K. (**2015**) Effects of cohesion on the structural and mechanical evolution of fold and thrust belts and contractional wedges: Discrete element simulations. *Journal of Geophysical Research: Solid Earth* 120:3870-3896. (可选, 涉及应力应变需引用此文)
- 有问题反馈

在使用中有任何问题, 欢迎反馈给李长圣。

- 邮件: sheng0619@163.com
- QQ: 836745132 (Neo)
- 官网: <https://geovbox.com>

第 13 章 版本更新

• V2.2 (2022-05-11)

1. 规则: 重新设置 `n_bond` 等 `bond` 参数, 将重置之前已经粘结的颗粒间的 `bond` 参数。例如

```
PROP n_bond 1.0 range id 1 2
PROP n_bond 2.0 range id 1 2
# 颗粒1和颗粒2的n_bond将被设置为2.0
# 之前的版本中, 颗粒1和颗粒2的n_bond不更新, 仍然保持为1.0
```

2. BUG 修复: 线弹性模型, `bond` 转为 `contact`, `kn` 和 `ks` 没有更新, 导致它们取值为 0。
3. BUG 修复: `sav 0000046.sav`, 缺失字符 0, 输出为 `000046.sav`。
4. BUG 修复: `zdem2jpg` 使用参数 `-bondplot=true` 时, `xmax ymax` 失效
5. 升级: 显著提升 `zdem2jpg` 计算速度,

• V2.1 (2021-06-17)

1. 规则: 墙 `wall` 默认为双面受力墙
2. BUG 修复: `zdem -s ./data` 多线程并行, 弹出错误 `mkdir err`
3. BUG 修复: `zdem2jpg` 使用 1 个核计算时内存溢出
4. BUG 修复: `zdem2jpg` 使用参数 `-wallshow=false` 时, `xmax ymax` 失效

第 14 章 致谢

南京大学徐雯峤、河北地质大学杨克基参与了本手册的修订工作，在此表示感谢。

感谢 [GMT 中文社区](#) 开源的网站及手册代码，对本站的建设提供了很大帮助。

感谢 [YADE](#) , [MatDEM](#) , [DICE2D](#) , [PFC2D/3D](#), [TRUBAL](#), [RICEBAL](#) 提供的离散元原理及代码开发方面的知识。

Thanks to [YADE](#), [MatDEM](#), [DICE2D](#), [PFC2D/3D](#), [TRUBAL](#), [RICEBAL](#), [gtkmm](#), [PLplot](#), [Cairo](#) and all the people who helped me.

Especially, we thank [Julia Morgan](#) for generously sharing her discrete element code [RICEBAL](#) (v. 5.4, modified from Peter Cundall' s [TRUBAL](#) v. 1.51), along with her post-processing scripts and algorithms, which have been used to process and display the model outputs presented in this website. Further details about these methods can be found in the following reference: [Morgan \(2015\)](#) (DOI: 10.1002/2014JB011455).

We would like to thank Thomas Fournier provided an open source MATLAB code for calculating stress and strain in his homepage hosted on the website of rice university, Chun LIU and Qian HUANG for discussions on development of ZDEM.

参考文献

- [Li2021] LI C.S., YIN H.W.*, et al. (2021) Calibration of the discrete element method and modelling of shortening experiments. *Front. Earth Sci.* 9:636512.
- [Zhao2015] Zhao G-F (2015) High performance computing and the discrete element model: opportunity and challenge. Elsevier,
- [Place1999] Place D, Mora P (1999) The lattice solid model to simulate the physics of rocks and earthquakes: incorporation of friction. *Journal of Computational Physics* 150:332-372(ISSN 0021-9991)
- [Liu2015] Liu C, Pollard DD, Gu K, Shi B (2015) Mechanism of formation of wiggly compaction bands in porous sandstone: 2. Numerical simulation using discrete element method. *Journal of Geophysical Research: Solid Earth* 120:8153-8168(ISSN 2169-9356)
- [Li2019] 李长圣 (2019) 基于离散元的褶皱冲断带构造变形定量分析与模拟. 南京大学
- [Cundall1979] Cundall PA, Strack OD (1979) A discrete numerical model for granular assemblies. *Geotechnique* 29:47-65. <https://doi.org/10.1680/geot.1979.29.1.47> (ISSN 0016-8505)

索引

B

ball, [52](#)

bond, [52](#)

C

convert, [52](#)

cyc, [53](#)

D

del, [53](#)

E

exp, [53](#)

F

fix, [54](#)

free, [54](#)

G

gen, [55](#)

gline, [56](#)

I

imple, [56](#)

init, [57](#)

L

load, [57](#)

P

prop, [57](#)

R

range, [59](#)

restore, [61](#)

S

set, [61](#)

start, [62](#)

W

wall, [62](#)

Z

zdem, [63](#)

zdem2jpg, [67](#)

zdemss, [64](#)