

# VBOX 手册 v1.4

李长圣

2021 年 04 月 11 日

保护环境, 从阅读电子文档开始!

# 目 录

第 1 章 简介	7	第 6 章 命令参考	39
第 2 章 Windows 下使用	9	第 7 章 离散元原理	55
第 3 章 一个示例学会 VBOX	13	第 8 章 颜色表	61
第 4 章 提交计算	17	第 9 章 开通账户	63
4.1 直接提交 . . . . .	17	第 10 章 Linux 命令行	65
4.2 LSF 调度 . . . . .	17	第 11 章 致谢	67
4.3 SGE 调度 . . . . .	18	第 12 章 使用协议	69
4.4 PBS 调度 . . . . .	19		
第 5 章 构造模拟	21		

保护环境, 从阅读电子文档开始!

欢迎来到 [VBOX](#) 的世界。

首先来看[一个示例学会 VBOX](#)

本项目是由 [VBOX](#) 官网维护的 VBOX 参考手册, 即可以作为 VBOX 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 VBOX 的使用方法。

### **VBOX 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。

### **VBOX 初级视频教程**

- [geovbox@ 哔哩哔哩](#)

### **相关链接:**

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

保护环境, 从阅读电子文档开始!

# 第 1 章 简介

- VBOX 是什么

VBOX，全称 Virtual sandBOX，即虚拟沙箱软件，是一个用于构造变形研究的二维离散元软件。始于 2014 年，采用 C 语言编写，并用 OpenMP 完成了并行设计，已获软件著作权。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。

- VBOX 的历史

- 2015 年，李长圣开发了 VBOX 的最原始版本 VBOX 1.0；
- 2018 年 8 月 10 日，VBOX 1.3 发布；
- 目前最新版本 VBOX 1.4 beta ( vboxdaily ) 发布于 2019-12-02。

- VBOX 开发者

VBOX 源码由 [李长圣](#) 开发并负责维护。



图 1.1: VBOX 的维护者

- VBOX 的特点

为什么选择 VBOX 作为构造模拟呢？

1. 免费

VBOX 是免费软件，托管在南京大学高性能计算中心，任何人均可使用。

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 linux 和 windows 操作系统, 官方提供 ubuntu14.04 和 windows7, windows10 支持。

#### 6. 模块化

VBOX 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

VBOX 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

#### • VBOX 替代品

在构造模拟方面, 还有一些软件也可以实现类似的功能, 可以作为 VBOX 的替代品。

1. PFC2D: [PFC2D](#)
2. Yade: [手册](#) | [代码下载](#)
3. MatDEM: [MatDEM](#)
4. DICE2D: [DICE2D](#)

---

**待处理:** 介绍更多 VBOX 替代品, 如 TRUBAL, RICEBAL 等。

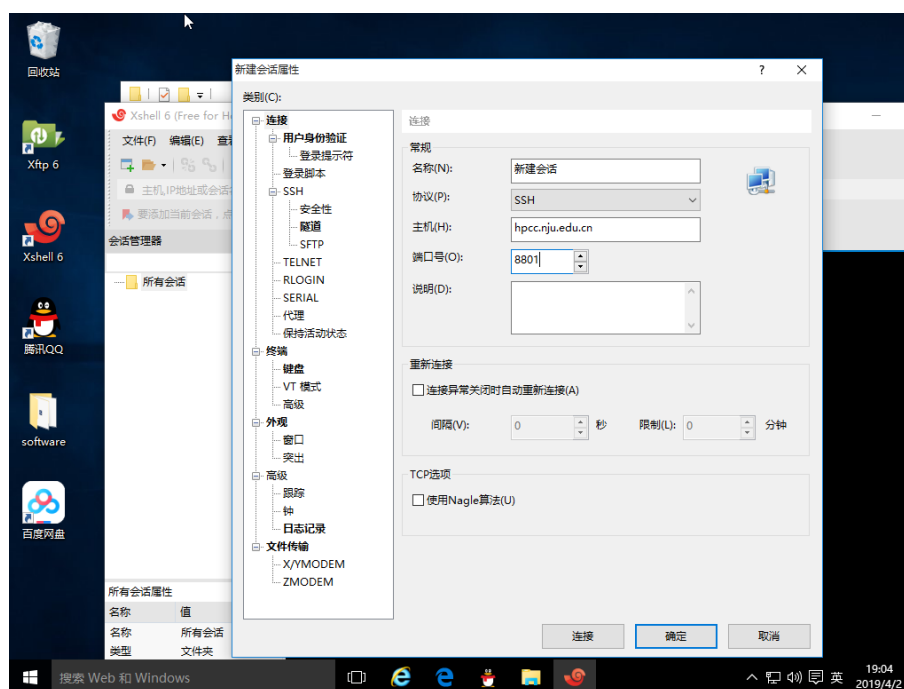
---



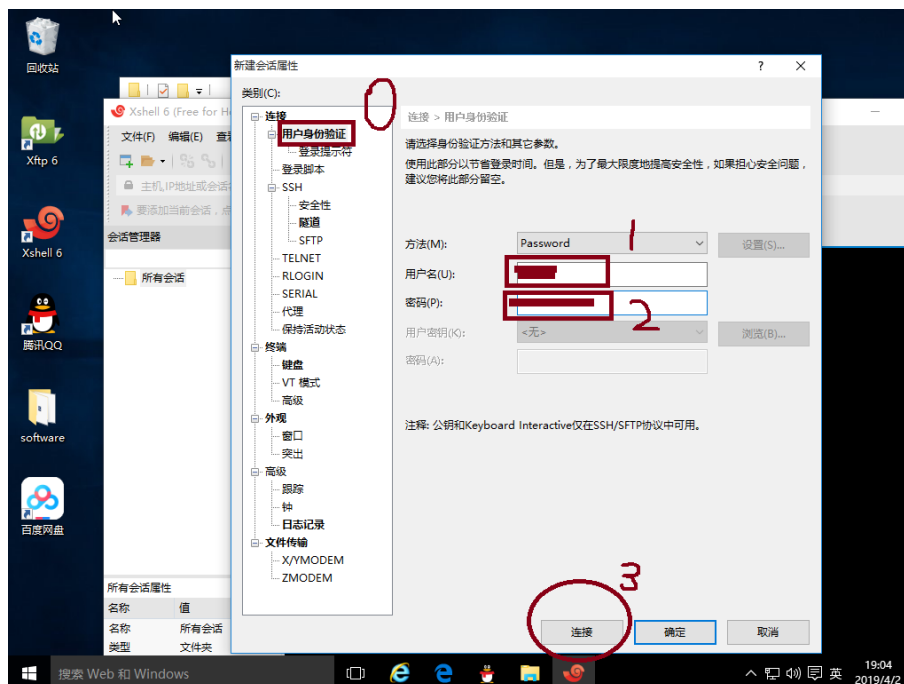
## 第 2 章 Windows 下使用

**注解：**小白用户请参考 [如何在南京大学高性能计算中心使用 VBOX](#)

1. 申请南京大学高性能计算中心帐号，申请办法见[开通账户](#)。
2. 下载安装 [Xshell](#)，使用 Xshell 连接南京大学高性能计算中心的主机。
  - 文件-新建
  - 连接南京大学高性能计算中心的主机 `hpcc.nju.edu.cn` 端口号 8801



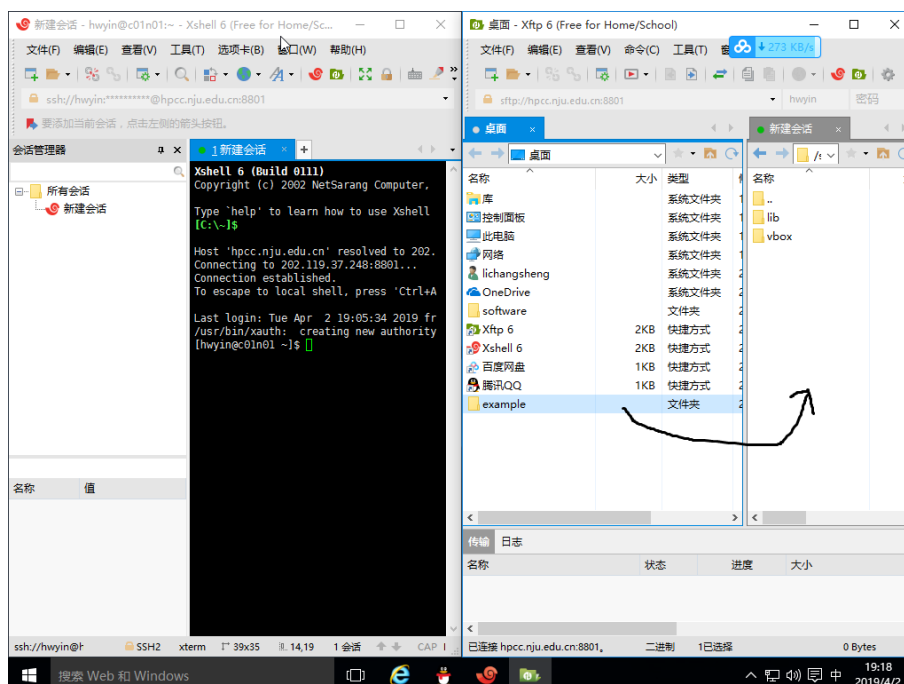
- 连接-用户身份验证填入南京大学高性能计算中心分配给你的用户名和密码，点击连接，登陆成功。



3. 下载安装 Xftp , 实现在主机 (南京大学高性能计算中心的集群) 和本地电脑间传输数据。下载 VBOX 的官方实例 example, 下载完, 利用 Xftp 将 example 传输到主机。

百度网盘 example 下载

链接: <https://pan.baidu.com/s/1bCYXzA3cITksu3jQRl8tCw> 提取码: 0h7r



4. 配置环境变量并, 运行示例 ex1\_push , 在 Xshell 中, 输入:

```
cd example # 进入 example 目录
chmod a+x ./vboxinstall.sh # 修改执行权限
./vboxinstall.sh # 安装 vbox
source ~/.bashrc # 更新环境变量
cd ex1_push # 进入 ex1_push 目录
vbox push.py # 开始计算
```

5. 计算完成后, 将 `ex1_push` 下载到本地, `ex1_push/data` 中保存了计算结果。可以使用 `gsview` 查看 ps 图片格式计算结果。

- 安装 `gsview`
  - [gsv50w32.exe](#) Win32 self extracting archive
  - [gsv50w64.exe](#) Win64 (x86\_64) self extracting archive
- 安装 Ghostscript
  - [Ghostscript 9.26 for Windows \(32 bit\)](#)
  - [Ghostscript 9.26 for Windows \(64 bit\)](#)

---

**待处理:** 如何使用 [paraview](#) 查看 vtk 格式计算结果

---

保护环境, 从阅读电子文档开始!

## 第 3 章 一个示例学会 VBOX

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 VBOX 的使用方法。

使用 Xshell 登陆南京大学高性能计算中心集群，运行 push.py 脚本：

```
vboxdaily push.py
```

push.py 中完整脚本命令如下

```
#####
# title: 一个实例学会 VBOX
# date: 2019-01-13
# 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  $4/3 \times \pi \times r^3$  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 fric0.0
→0.0 COLOR black
WALL ID 1, NODES ( 10.0 , 10000.0 ) ( 10.0 , 10.0 ), kn=0e10 ks=0e10 fric0.0
→0.0 COLOR blue
WALL ID 2, NODES ( 40000.0 , 10.0 ) ( 40000.0 , 10000.0 ), kn=0e10 ks=0e10 fric0.0
→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 ), COLOR0.0
→black GROUP ball_rand
# 设置颗粒的微观参数 density 密度, firc 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
```

(下页继续)

(续上页)

```

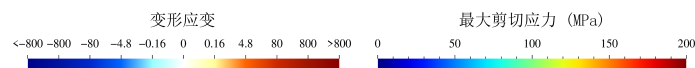
PROP density 2.5e3, fric 0.0, shear 2.9e9, poiss 0.2, damp 0.4, hertz
# 设置时间步及重力加速度
SET DT 5e-2, GRAVITY 0.0, -9.8
# 设置每 1000 步保存一次 vtk 格式的计算结果
SET vtk 1000
# 设置每 1000 步保存一次 ps 格式的计算结果
SET ps 1000
# 设置每 1000 步保存一次 dat 格式的计算结果
SET print 1000
# 沉积, 计算 5000 步
CYC 5000
# 删除 4000 米以上的颗粒
DEL RANGE y 4000.0 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:



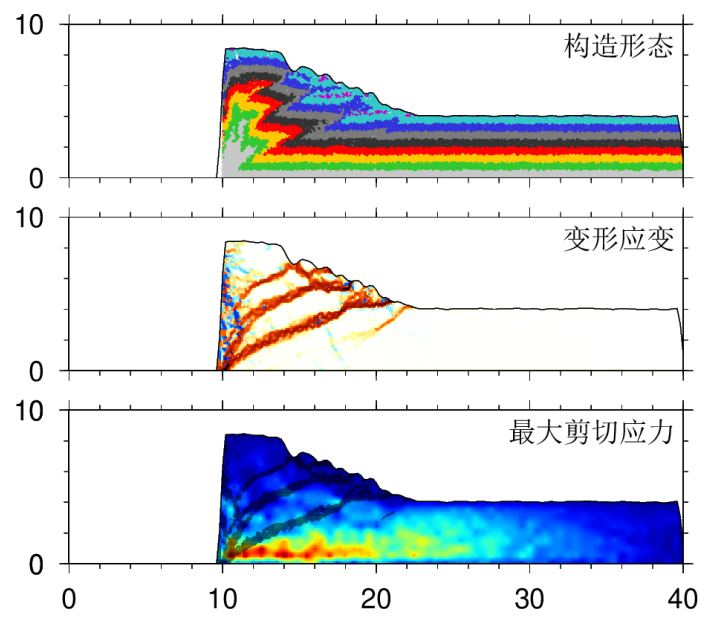


图 3.1: 单位 (km)

颗粒参数表

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

保护环境, 从阅读电子文档开始!



## 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  
vbox 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 `xshell`, 计算停止

### 4.2 LSF 调度

南京大学高性能计算中心安装了该调度系统。

- `bsub < lsf.sh` 提交
- `bjobs` 查看
- `bkill <id>` 删除
- 优点: 同时计算多个, 关闭 `xshell`, 不影响计算
- 缺点: 不能实时查看计算情况

#### 1. 提交计算

```
bsub < lsf.sh
```

显示 Job <230259> is submitted to queue <mpi>. 说明提交成功。

lsf.sh 中内容如下:

```
#!/bin/bash  
#  
#BSUB -J sheng  
#BSUB -q mpi  
#BSUB -n 24  
#BSUB -R "span[ptile=24]"  
vboxdaily push.py  
vbox2jpg --dir=./data # 生成 jpg  
convert -delay 100 ./data/*[0-9].jpg -loop 0 ./data/process.gif # 制作  
↪gif
```

#### 2. 查看计算状态

```
bjobs
```

显示

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	
→SUBMIT_TIME							
230259	hwyin	RUN	mpi	c01n02	24*c38n11	*push.py	Apr 10 12:44
→12:44							

从上面的输出可知, 该任务的 id 为 230259。STAT 状态为 RUN 说明正在计算。任务分配给了节点 c38n11, 调用了 24 个核计算。更为详细的说明, 可以查询 LSF 调度系统教程。其中, `-J` 参数为任务名, 可自定义。

### 3. 强制结束计算

```
bkill <id>
```

我们要杀死上面的任务, `bkill 230259` 即可以。输出

```
Job <230259> is being terminated
```

说明任务已经被杀死。

## 4.3 SGE 调度

南京大学尹宏伟课题组和东华理工大学并行计算实验室集群安装了该调度系统

- `qsub sge.sh` 提交
- `qstat` 查看
- `qdel <id>` 删除
- 优点: 同时计算多个, 关闭 `xshell`, 不影响计算
- 缺点: 不能实时查看计算情况

### 1. 提交计算

```
qsub sge.sh
```

显示 Your job 3136 ("example") has been submitted 说明提交成功。

`sge.sh` 中内容如下:

```
#!/bin/sh
#$ -S /bin/bash
#$ -N example # 设置任务的名字
#$ -j y
#$ -M sheng@163.com # 修改为自己邮箱
```

(下页继续)

(续上页)

```

#$ -m e
#$ -o log.txt # 程序执行日志, 记录了错误信息
#$ -V
#$ -cwd
#$ -pe orte 8 # 调用 8 个核计算
export OMP_NUM_THREADS=$NSLOTS
time vboxdaily push.py # 提交计算
vbox2jpg --dir=./data # 生成 jpg
convert -delay 100 ./data/*[0-9].jpg -loop 0 ./data/process.gif # 制作
→gif

```

## 2. 查看计算状态

```
qstat
```

显示

job-ID	prior	name	user	state	submit/start at	queue
-----						
3136	0.55500	example	zhangsan	r	01/20/2019 14:43:53	all.
→q@sand-0-0.local			8			

从上面的输出可知, 该任务的 id 为 3136。state 状态为 r 说明正在计算, 如果为 w 为排队等待状态。任务分配给了节点 sand-0-0, 调用了 16 个核计算。更为详细的说明, 可以查询 SGE 调度系统教程。

## 3. 强制结束计算

```
qdel <id>
```

我们要杀死上面的任务, qdel 3136 即可以。输出

```
zhangsan has registered the job 3136 for deletion
```

说明任务已经被杀死。

## 4.4 PBS 调度

中国石油勘探开发研究院 (北京) 集群安装了该调度系统

PBS(Portable Batch System) 最初由 NASA 的 Ames 研究中心开发, 主要为了提供一个能满足异构计算网络需要的软件包, 用于灵活的批处理, 特别是满足高性能计算的需要, 如集群系统、超级计算机和大规模并行系统。PBS 的主要特点有: 代码开放, 免费获取; 支持批处理、交互式作业和串行、多种并行作业。

### 1. qsub pbs.sh 提交

```
[zhangsan@mu01 ~]$ qsub pbs.sh
28.mu01
```

其中, `pbs.sh` 中内容:

```
#PBS -N sheng ## 可以给任务一个名字, 方便辨识
#PBS -l nodes=1:ppn=12 ## 使用 1 节点, 每个节点 12 核
cd $PBS_O_WORKDIR ## 到当前工作目录下 (此为 PBS 提供的环境变量)
vboxdaily push.py
vbox2jpg --dir=./data
convert -delay 100 ./data/*[0-9].jpg -loop 0 ./data/process.gif
```

## 2. qstat 查看

```
[zhangsan@mu01 ~]$ qstat
Job id          Name          User          Time Use S
--
Queue
-----
--
31.mu01         zhangsan      lichangsheng  00:00:13 R
batch
```

## 3. qdel 31 删除作业号为 31 的作业

```
[zhangsan@mu01 ~]$ qdel 28
[zhangsan@mu01 ~]$ qstat
Job id          Name          User          Time Use S
--
Queue
-----
--
31.mu01         sheng         zhangsan      00:22:43 C batch
```

## 参数解释

Job id	Name	User	Time Use	S	Queue
作业号	作业名	提交人	运行的时间	作业状态	队列

通常作业状态:

- Q 作业排队
- R 作业执行
- C 作业清除
- E 作业退出
- H 作业挂起

## 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

### 同构造剥蚀

这里是一个 同构造剥蚀计算实例。

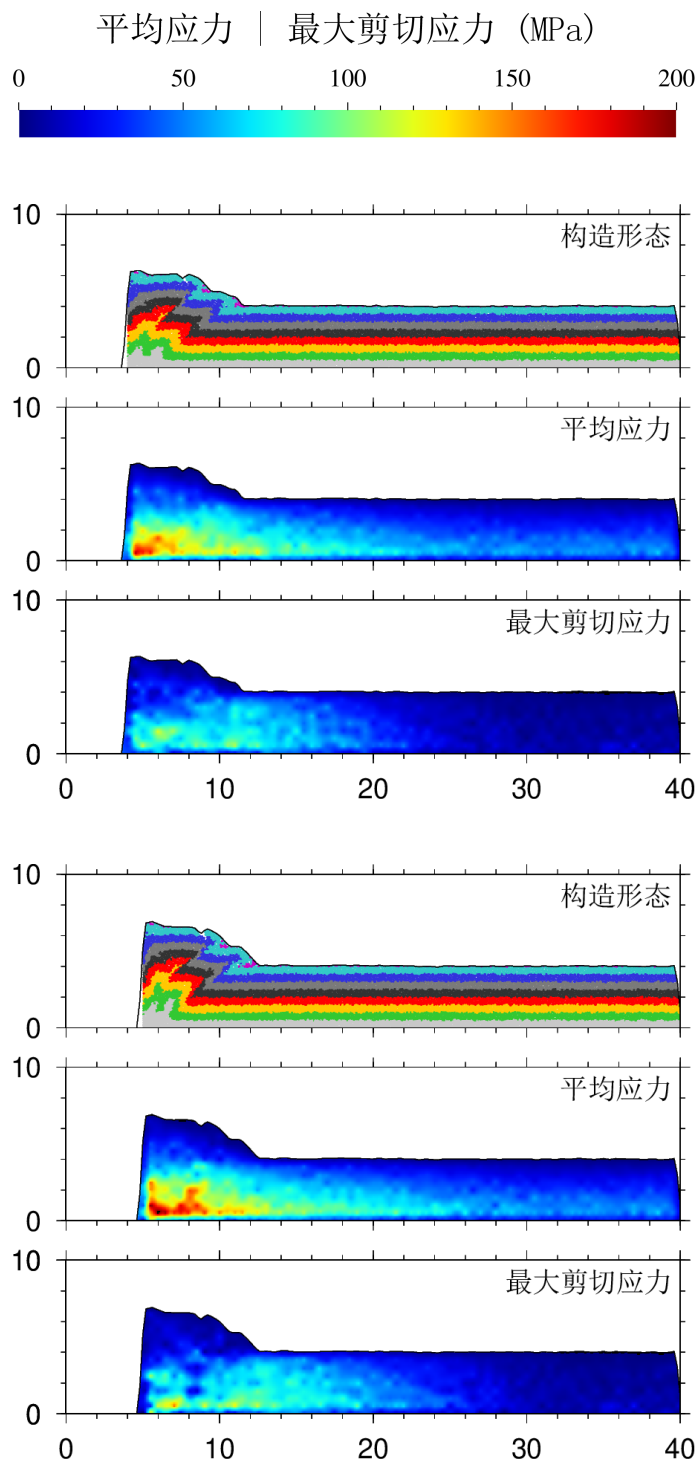
登陆南京大学高性能计算中心集群, 运行 `syn_erosion.py` 脚本:

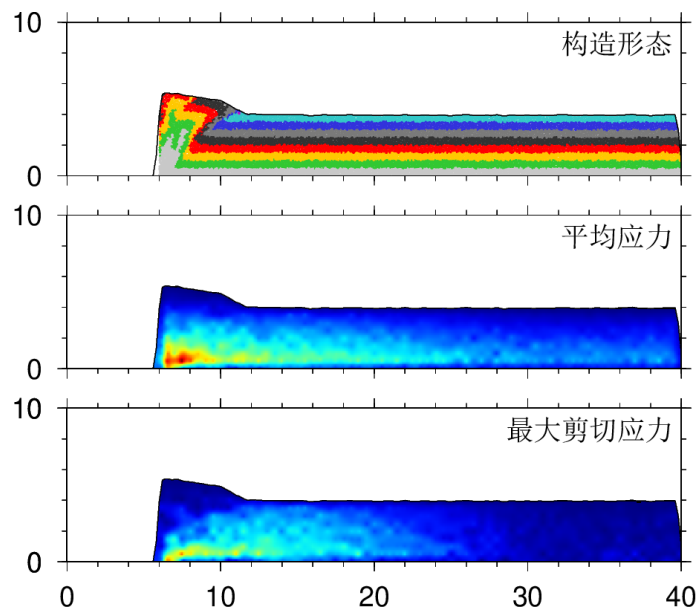
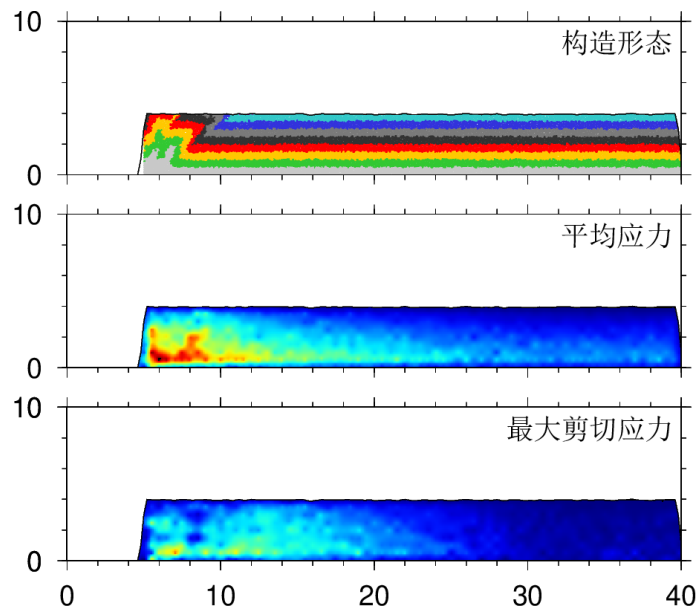
```
vbox syn_erosion.py
```

`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 VBOX 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:





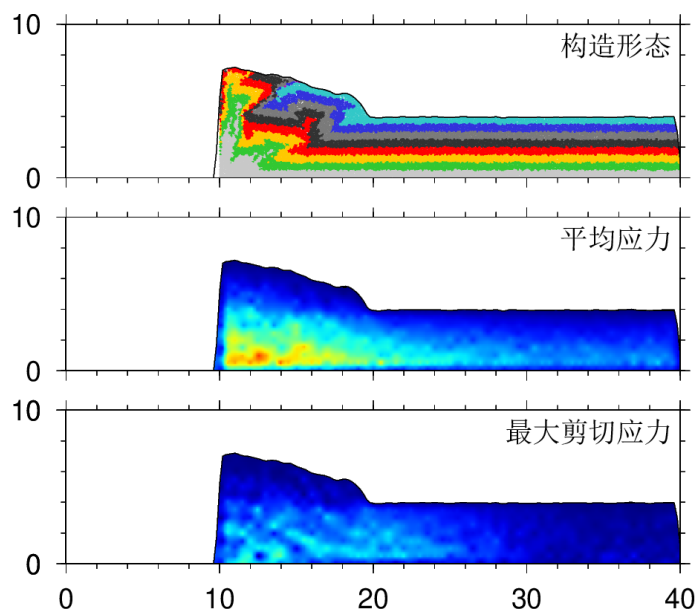


图 5.1: 单位 (km)

## 同构造沉积

这里是一个 同构造沉积计算实例。

登陆南京大学高性能计算中心集群, 运行 `syn_sedimentaion.py` 脚本:

```
vbox syn_sedimentaion.py
```

`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.geovbox.com
#####
# 从 0000046000.sav < 版本 1.3> 计算节点恢复, 由 一个实例学会 VBOX 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 4000~6000.0.
# 经验: 颗粒充填满 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 ( 12000.0, 40000.0), y ( 4000.0, 6000.0),
COLOR red GROUP sed
```

(下页继续)



(续上页)

```

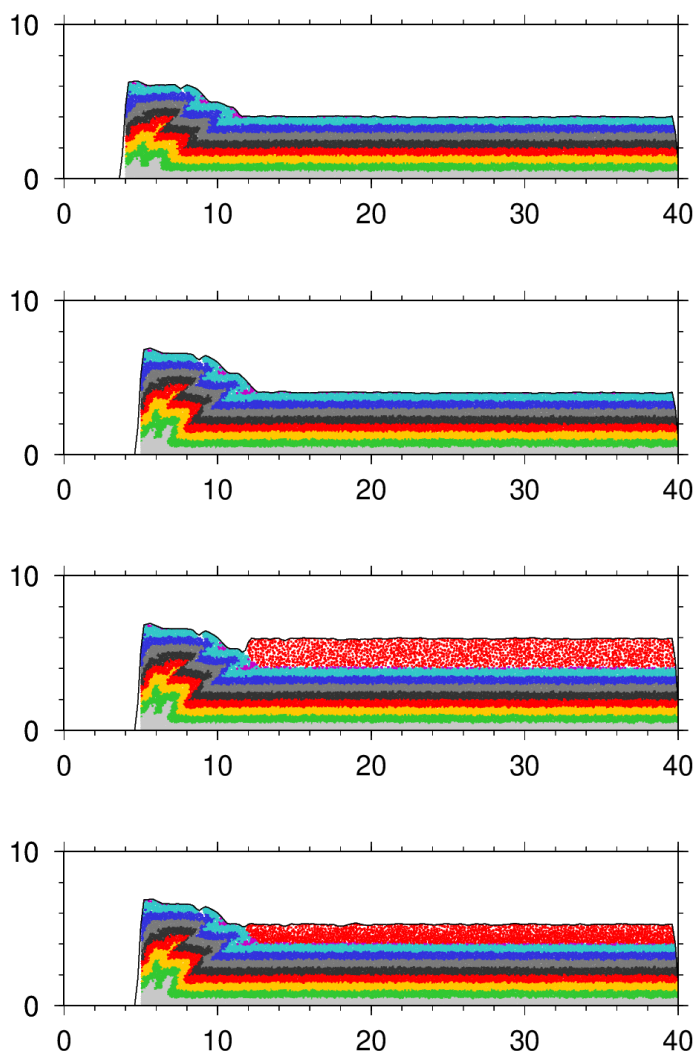
# 设置沉积颗粒 GROUP=sed 的微观参数
PROP DENSITY 2.5e3, fric 0.3, shear 2.9e9, poiss 0.2, damp 0.4, hertz range GROUP sed
# 计算 2000 步, 让颗粒沉积下来
SET print 100 # 每 100 步输出一次计算结果
CYC 2000
#####

# 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:



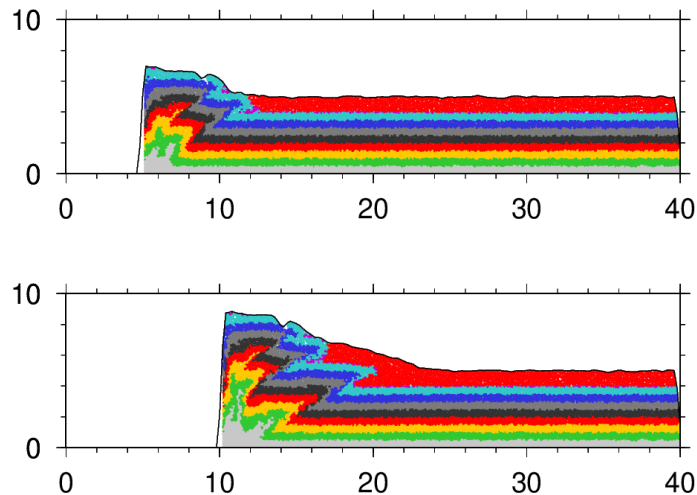


图 5.2: 单位 (km)

## 先存断层设置

这里是一个 断层设置实例。

登陆南京大学高性能计算中心集群, 运行 `pre_struct.py` 脚本:

```
vbox pre_struct.py
```

`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  $4/3 \times \pi \times r^3$  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 0.00
↪0.0 COLOR black
WALL ID 1, NODES ( 10.0 , 10000.0 ) ( 10.0 , 10.0 ), kn=0e10 ks=0e10 fric 0.00
↪0.0 COLOR blue
WALL ID 2, NODES ( 40000.0 , 10.0 ) ( 40000.0 , 10000.0 ), kn=0e10 ks=0e10 fric 0.00
↪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 0.00
↪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
```

(下页继续)

(续上页)

```

# 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
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:

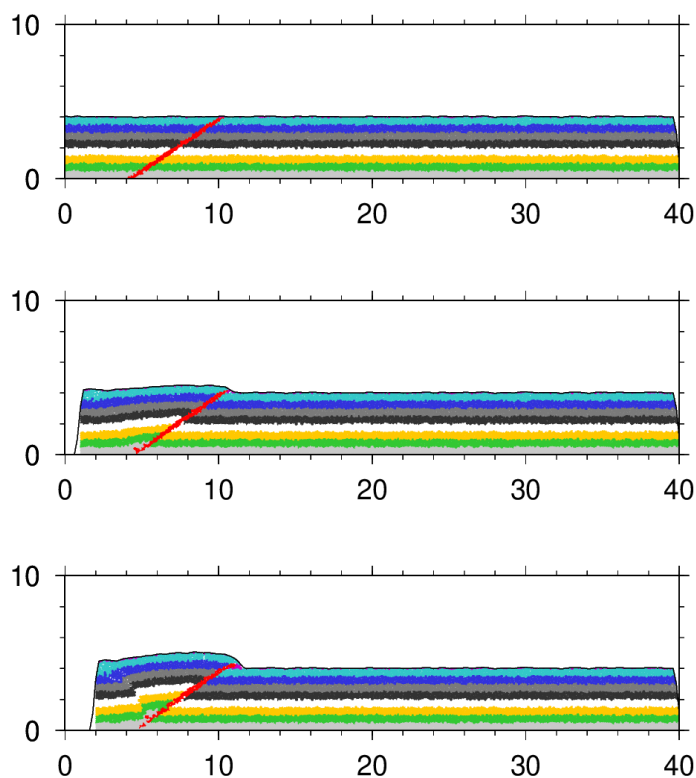


图 5.3: 单位 (km)

## 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。

ex5 目录中文件有

- [lsf.sh](#)
- [detachment.py](#)
- [init\\_xyr.dat](#)

## 进入 ex5 目录, 在终端或 xshell 中运行

```
$ bsub < lsf.sh # 提交计算
$ bjobs        # 查看计算是否提交成功
```

lsf.sh 中内容如下:

```
#!/bin/bash
#
#BSUB -q mpi
#BSUB -n 24
#BSUB -R "span[ptile=24]"
vboxdaily detachment.py
vbox2jpg --dir=./data/ --xmax=40000 --ymax=10000 # gen jpg
#convert -delay 100 ./data/*[0-9].jpg -loop 0 ./data/process.gif # gen gif
```

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.geovbox.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
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, 摩擦系数可以根据断层强弱改变
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
```

(下页继续)

(续上页)

```
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:

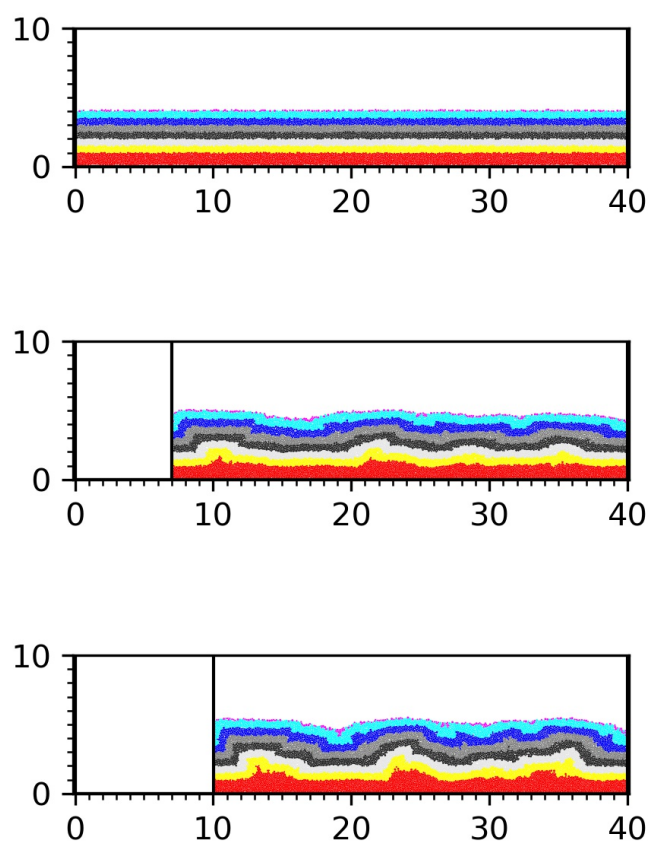


图 5.4: 单位 (km)

#### 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.

## 古隆起

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

ex6 目录中文件有

- [lsf.sh](#)
- [palaeohigh.py](#)
- [init\\_xyr.dat](#)

## 进入 ex6 目录, 在终端或 *xshell* 中运行

```
$ bsub < lsf.sh    # 提交计算
$ bjobs           # 查看计算是否提交成功
```

lsf.sh 中内容如下:

```
#!/bin/bash
#
#BSUB -q mpi
#BSUB -n 24
#BSUB -R "span[ptile=24]"
vboxdaily palaeohigh.py
vbox2jpg --dir=./data/ --xmax=40000 --ymax=10000 # gen jpg
#convert -delay 100 ./data/*[0-9].jpg -loop 0 ./data/process.gif # gen gif
```

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 0.00
→0.0 COLOR black
WALL ID 1, NODES ( 10.0 , 10000.0 ) ( 10.0 , 10.0 ), kn=0e10 ks=0e10 fric 0.00
→0.0 COLOR blue
WALL ID 2, NODES ( 40000.0 , 10.0 ) ( 40000.0 , 10000.0 ), kn=0e10 ks=0e10 fric 0.00
→0.0 COLOR red
# 设置颗粒的微观参数
```

(下页继续)

```

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)

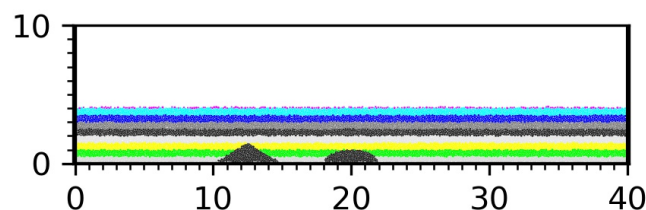
# 设置 bond 粘结, 使颗粒具有粘聚力
PROP ebmod 2e8 gbmod 2e8 tstrength 0e7 sstrength 0e7 fric 0.3
# 给地层赋上颜色
PROP COLOR lg      range y    0.0    500.0
PROP COLOR green   range y   500.0  1000.0
PROP COLOR yellow  range y  1000.0  1500.0
PROP COLOR white   range y  1500.0  2000.0
PROP COLOR black   range y  2000.0  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(2000.0, 0.0) 长轴 2000.0 短轴 1000.0
PROP GROUP palaeohigh RANGE ellipse ( 2000.0, 0.0) 2000.0, 1000.0
# 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 struct1 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:





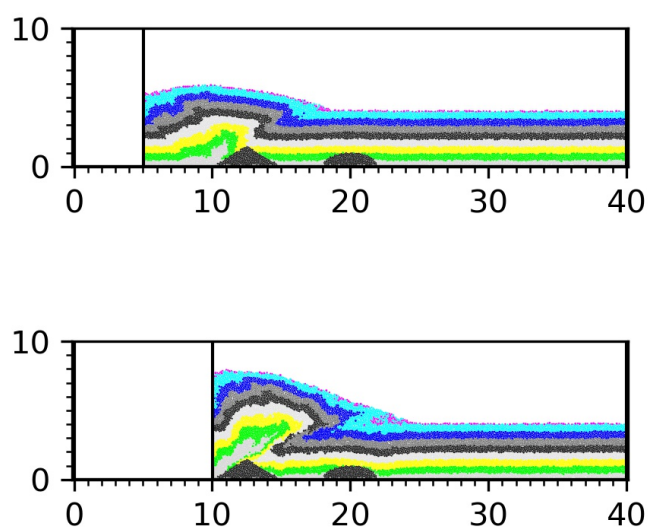


图 5.5: 单位 (km)

## 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。

ex7 目录中文件有

- [lsf.sh](#)
- [extens\\_rigid.py](#)
- [init\\_xyr.dat](#)

## 进入 ex7 目录, 在终端或 xshell 中运行

```
$ bsub < lsf.sh # 提交计算
$ bjobs        # 查看计算是否提交成功
```

lsf.sh 中内容如下:

```
#!/bin/bash
#
#BSUB -q mpi
#BSUB -n 24
#BSUB -R "span[ptile=24]"
vboxdaily extens_rigid.py
vbox2jpg --dir=./data/ # gen jpg
#convert -delay 100 ./data/*[0-9].jpg -loop 0 ./data/process.gif # gen gif
```

extens\_rigid.py 中内容如下:

```
#2019-06-29
# 建议使用 notepad++ 查看本文件
#LI ChangSheng @ NanJing Uninversity
```

(下页继续)

```

#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
##### 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

```

(续上页)

```

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
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 ballsed3
PROP color mg den 2.5e3, fric 0.3, shear 2.9e9, poiss 0.2, damp 0.0, hertz range group
→ballsed3
;prop ebmod 2e8 gbmod 2e8 tstrength 2e7 sstrength 4e7 fric 0.3 range group ball_rand
INI XV 0.0 RANGE GROUP bom_wall2
WALL ID 1 XV 0.0
set ps 1000

```

(下页继续)

(续上页)

```

set print 1000
CYC 5000
DEL RANGE y 11000.0 16000.0
CYC 2000

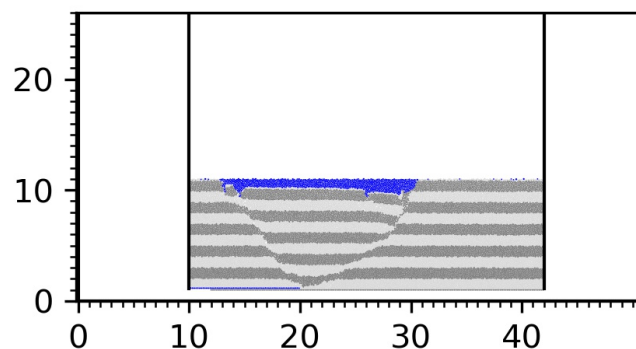
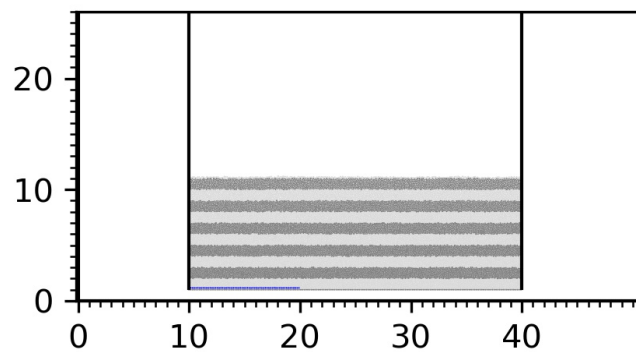
# 开始伸展
INI  XV 2.0 RANGE GROUP bom_wall2
wall id 1 xv 2.000
IMPLE WALL ID 1 XMOVE 2000.0 SAVE 2000.0 PRINT 1000.0 PS 1000.0

##### 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
;prop ebmod 2e8 gbmod 2e8 tstrength 2e7 sstrength 4e7 fric 0.3 range group ball_rand
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:



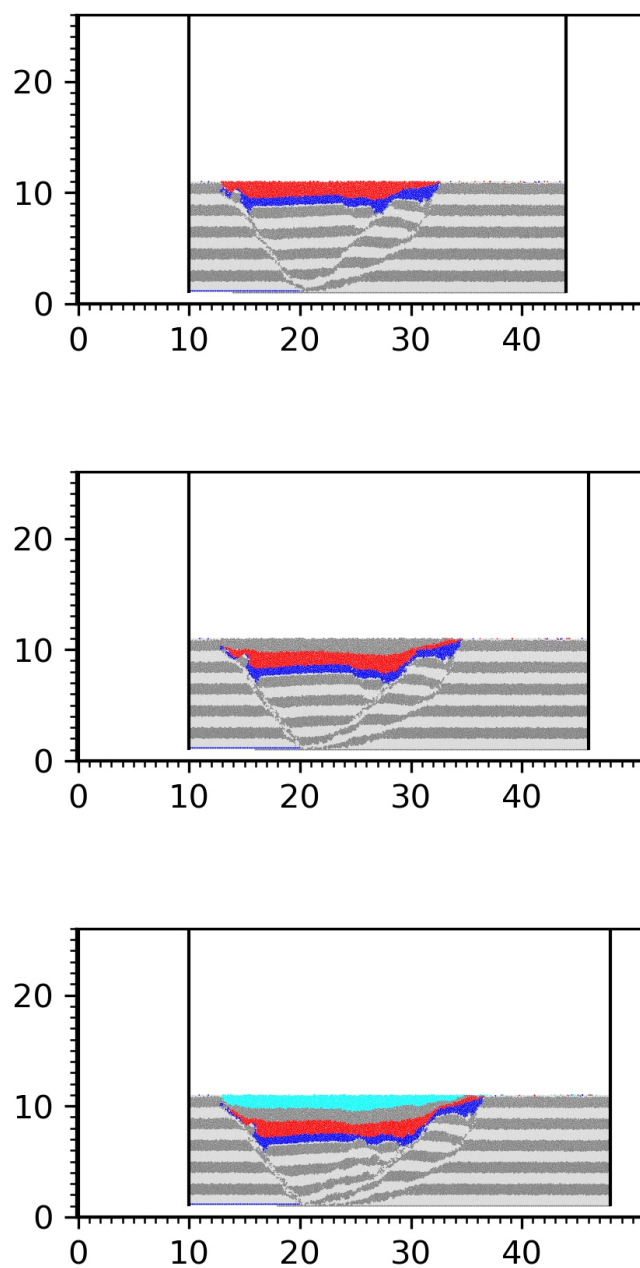


图 5.6: 单位 (km)

## 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.

保护环境, 从阅读电子文档开始!

## 第 6 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[l] id=<int> X=<float> Y=<float> :`

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

### 按功能分类

下面将 VBOX 中的命令按照功能分类, 并用一句话简述其功能。

- 主程序
  - `vbox` : `vbox push.py`, 运行 `push.py` 脚本
- 格式转换、gif 制作
  - `vbox2jpg` : 将计算结果绘制成 jpg 格式, 同 `vboxplot`
  - `gmt psconvert` : 将 VBOX 生成的 PS 文件转换为其他图片格式。`GMT` 命令
  - `convert` : 制作 gif

脚本 `cmd.txt` 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
  - *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[l]* 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 VBOX 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

- 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
```

- 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
```

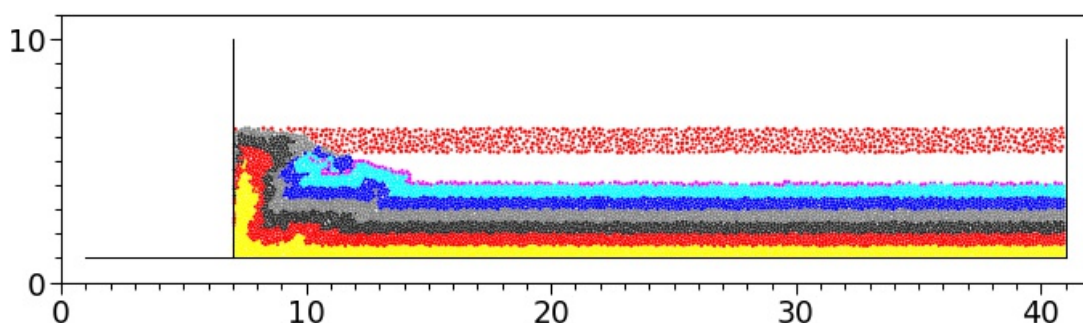


图 6.1: 沉积前

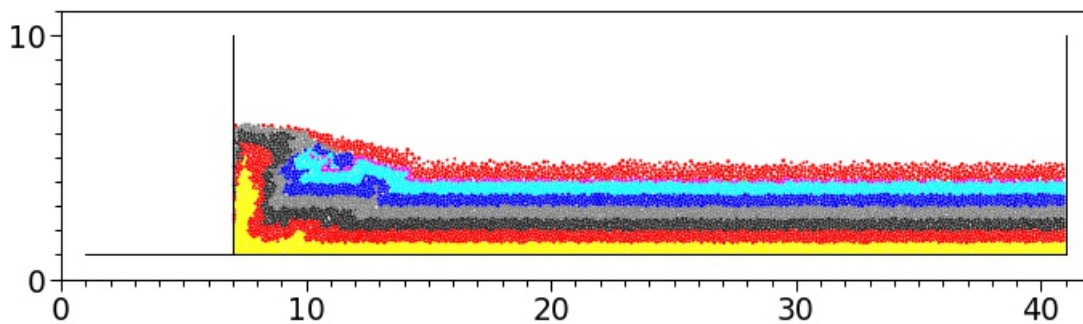


图 6.2: 沉积后

## gline

说明 `gli[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 为初始叠和率,  $cratio = |OA|/(rO+rA)$ , 取值范围 (0.0~1.0)。当颗粒刚好接触时,  $cratio=(rO+rA)/(rO+rA)=1.0$ ; 当颗粒完全重合时,  $cratio=0.0/(rO+rA)=0.0$

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| = (rO+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 `imp[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
```

## 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 VBOX 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

### 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 ...]
```

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
# 设置所有颗粒粘结在一起, 使颗粒具有粘聚力
```

(下页继续)

(续上页)

```
PROP n_bond 1.0e4 s_bond 1.0e4
#
#
# 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
```

## psconvert

**简介** 将 VBOX 生成的 PS/EPS 文件转换为其他图片格式

该模块通过调用 GhostScript 将 PS/EPS 文件转换为其他图片格式, 包括 BMP、EPS、JPEG、PDF、PNG、PPM、SVG、TIFF 格式。

### 必选选项

**psfiles** 要转换格式的 PS 文件名

默认情况下, 转换后的文件与转换前的文件有相同的文件名, 文件后缀由文件格式决定。

如下命令会将 PS 文件转换成 JPG 格式:

```
gmt psconvert test.ps
gmt psconvert test1.ps test2.ps map*.ps
```

### 可选选项

**-A** 对输出的图片做裁边

默认情况下, 转换得到的图片的大小由 PS 文件的纸张尺寸决定。通常画图的时候是不会把一张 A4 纸画满的, 所以在图片周围就会出现多余的白色部分。

-A 选项会对 PS 文件进行裁剪, 仅保留其中有绘图的部分:

```
gmt psconvert -A test.ps
```

**-D<outdir>** 设置输出目录

默认情况下, 会在 PS 文件同一目录中生成其他图片文件, 使用 -D<outdir> 选项可以指定输出目录, -D. 表示在当前目录输出。需注意, 输出目录必须已存在, 否则会报错。

**-E<dpi>** 设置图片精度

值越大, 图片越清晰, 文件也越大。PDF 格式默认值为 720, 其他格式默认值为 300, 单位为 dpi。

```
gmt psconvert -Tj -E600 test.ps
```

说明:

1. EPS 是矢量图片格式, -E 选项对其无效
2. PDF 是矢量图片格式, -E 选项对其中的 pattern 和字体有效

**-Tb|e|E|f|F|j|g|G|m|s|t** 指定要转换的图片格式。可以接如下值:

- b : BMP;
- e : EPS;
- E : 带有 PageSize 命令的 EPS;
- f : PDF;
- F : 多页 PDF;
- j : JPEG (默认值);
- g : PNG;
- G : 透明 PNG;
- m : PPM;
- s : SVG;
- t : TIFF;

说明:

1. g 和 G 的区别在于前者背景色为白色, 后者背景色为透明;
2. 对于 **bjgt** 格式可以在其后加 - 将 PS 文件转换为灰度图;
3. EPS 格式可以与其他格式合在一起使用。比如 **-Tef** 会同时生成 EPS 和 PDF 文件。除此之外, 该命令一次只能转换一种格式, 比如 **-Tbf** 则只会生成 PDF 格式 (即以 -T 选项中的最后一个格式为准)
4. **-TF** 会将多个 PS/PDF 文件转换并合并成一个多页的 PDF 文件, 需要使用 **-F** 选项指定输出的文件名

转换为 PDF 格式:

```
gmt psconvert -Tf test.ps
```

转换为 JPG 格式的灰度图:

```
gmt psconvert -Tj test.ps
```

利用一堆 PS 文件生成一个多页 PDF:

```
gmt psconvert -TF -Fout.pdf *.ps
```

注解: psconvert 详解见 <https://docs.gmt-china.org/6.0.0/module/psconvert>

## 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

```
# 圈定一个椭圆范围, 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}
```



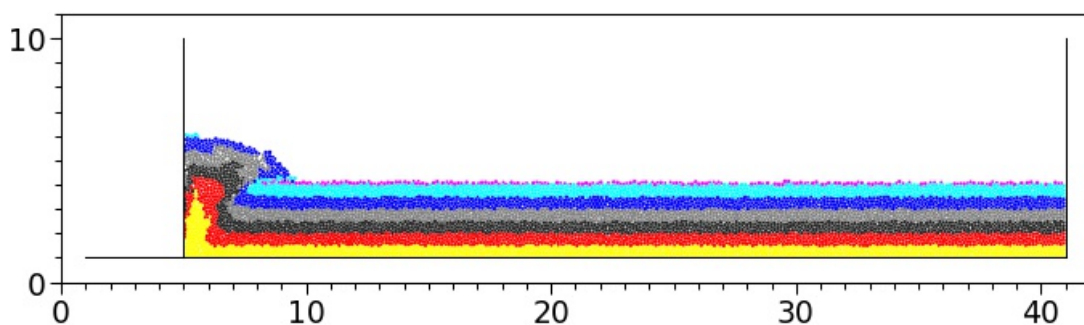


图 6.3: 剥蚀前

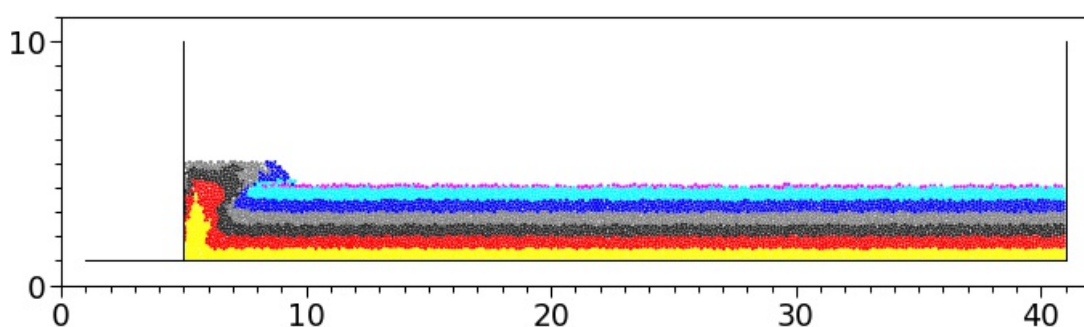


图 6.4: 剥蚀后

## 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 color 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, 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 VBOX 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>
```

设置进度条刷新间隔:

```
SET stepbar <int>
```

设置 VBOX 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 VBOX 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>, [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
```

## vbox

说明 VBOX 主程序。

使用方法

`vbox` [可选项] [script.py]

其中, `script.py` 为脚本名, 如 `vbox script.py`, 运行脚本 `script.py`, `cmd.txt` 为 ASCII 码格式文件。

## 可选项

- g, --grid SIZE (完善中...)** 配合-s 选项, 设置应力应变计算时候, 网格的大小, 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。
- s, --strin-stres DataDir (完善中...)** 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

## 实例

vbox -v

```
*****
Virtual SandBox(2D) version 1.4(2020-03-19) 教育版
官网: www.geovbox.com
邮箱: sheng0619@163.com
Copyright © 南京大学, 2014/2019
Copyright © VBOX, 2019/2020
*****
```

vbox -h

vbox -j 8 script.py 使用 8 个线程并行计算

vbox -s ./data 读取 ./data 中的数据计算应力应变

## vbox2jpg

---

注解: 同 vboxplot

---

**说明** 读取 VBOX 生成的 all\_\*.dat 文件绘制如 jpg 格式的图片

## 实例

```
vbox2jpg --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/zhangsan/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 vbox2jpg --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 vbox2jpg --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

保护环境, 从阅读电子文档开始!

## 第 7 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。

### 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;
    U_Ball_Steel=U_Ball_Steel+V_Ball_Steel*dT; %update displacement
```

(下页继续)

(续上页)

```

U_Ball_Wood = U_Ball_Wood+ V_Ball_Wood*dT;
Y_Ball_Steel = Y0_Ball_Steel + U_Ball_Steel; %update position
Y_Ball_Wood = Y0_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。

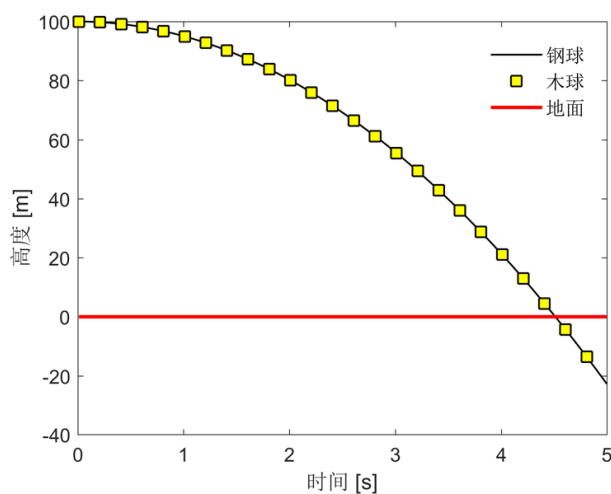


图 7.1: 球高度随时间的变化, 比萨斜塔离散元模拟试验。

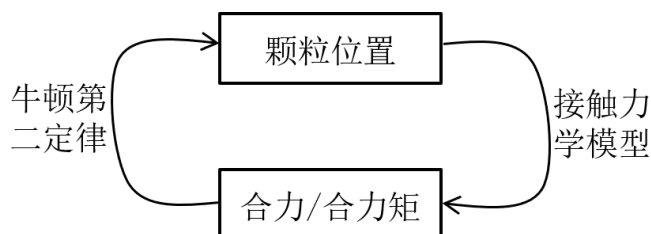


图 7.2: 球高度随时间的变化, 比萨斜塔离散元模拟试验。

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,  $F\_Ball\_Steel = -G * M\_Ball\_Steel$ , 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。

## 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  $\eta_t$  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  $\eta_t$  时的位置  $\vec{P}_t$  和  $t - \Delta t$  平动加速度  $\vec{v}_{t-\Delta t/2}$  时的平动速度, 则按照跳蛙法有,

$$\vec{v}_{t+\Delta t/2} = \vec{v}_{t-\Delta t/2} + \vec{a}_t \cdot \Delta t$$

$$\vec{P}_{t+\Delta t} = \vec{P}_t + \vec{v}_{t+\Delta t/2} \cdot \Delta t$$

其中,  $\Delta t$  为一个时步。

首先, 得到颗粒在  $t + \Delta t/2$  时的速度  $\vec{v}_{t+\Delta t/2}$ , 再得到颗粒在  $t + \Delta t$  时的位置  $\vec{P}_{t+\Delta t}$ 。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  $\theta_t$  和角加速度为  $\eta_t$ ,  $t - \Delta t/2$  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。

## 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)。

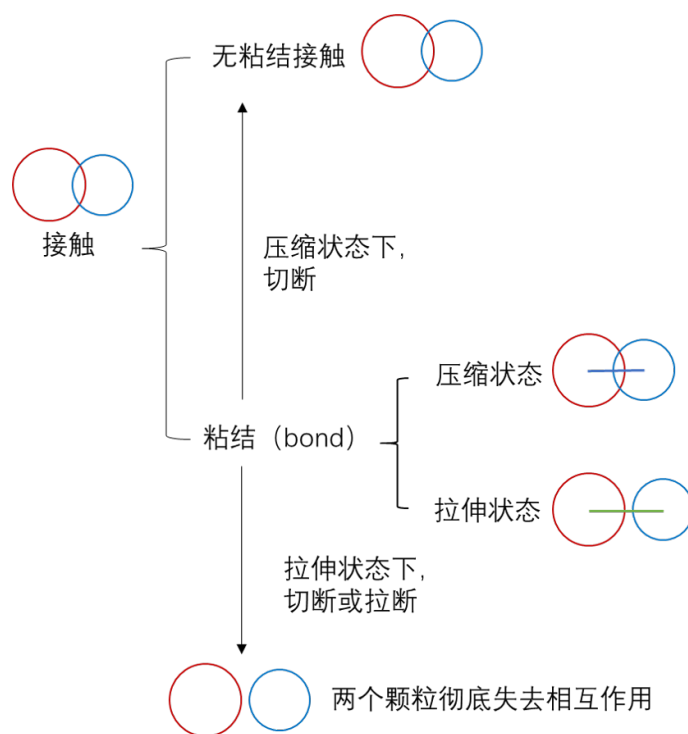


图 7.3: 接触与粘结关系

本文涉及的接触模型分成两类线弹性模型和 Hertz-Mindlin 模型。

## 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]

## 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]

保护环境, 从阅读电子文档开始!

## 第 8 章 颜色表

VBOX 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 `vbox2jpg` 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 9 章 开通账户

使用 VBOX 需要遵守用户协议, 见[使用协议](#)。

由于 VBOX 业务升级, 现暂停南京大学高性能计算中心 VBOX 账户申请。已申请南京大学高性能计算中心 VBOX 账户的用户, 遵守[使用协议](#) (2020-09-15), 可以正常使用软件, 不会受到任何影响。

保护环境, 从阅读电子文档开始!



## 第 10 章 Linux 命令行

使用 VBOX 前, 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
```

---

**注解:** 更详细的教程请参考 [Linux 命令行基本操作知识](#)。

---

## 第 11 章 致谢

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 VBOX.

保护环境, 从阅读电子文档开始!

# 第 12 章 使用协议

## VBOX1.4 教育版用户使用协议教育版

2020-09-15 用户使用 VBOX1.4 教育版, 默认同意本协议。

VBOX, 全称 Virtual sandBOX, 即虚拟沙箱软件, 是一个用于构造变形研究的二维离散元软件。VBOX 采用 C 语言编写, 基于 OpenMP 实现并行计算, 2015 年申请了软件著作权, 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。

VBOX1.4 教育版免费提供给用户使用, 用户仅可作教育、科研使用, 不得将软件用于商业用途。用户 发表论文、作会议报告时, 须注明数值模拟使用了 VBOX, 并按照以下示例致谢并引用相关文献。发表文章时, 致谢南京大学高性能计算中心, 南京大学高性能计算中心同时会制定新的文章奖励政策。

- 致谢示例

- 中文版

本文的数值计算是在南京大学高性能计算中心的计算集群上完成的, 模拟实验使用的软件为东华理工大学李长圣研发的离散元数值模拟软件 VBOX([www.geovbox.com](http://www.geovbox.com)), 在此表示感谢。

- 英文版 1(只含南京大学高性能计算中心, 其余请自己翻译)

We are grateful to the High Performance Computing Center (HPCC) of [Nanjing University](#) for doing the numerical calculations in this paper on its blade cluster [system](#).

- 英文版 2(只含南京大学高性能计算中心, 其余请自己翻译)

The numerical calculations in this paper have been done on the computing facilities in the High Performance Computing Center (HPCC) of Nanjing University.

- 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. 李长圣, 尹宏伟 \*, 刘春, 蔡申阳. 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), 2017(06): 1161-1170. (可选, 偏重软件开发)

3. LI Changsheng, YIN Hongwei\*, 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. 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. (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](mailto:sheng0619@163.com)
- QQ: 836745132 (Neo)
- VBOX 官网: <https://geovbox.com>

## 参考文献

- [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, [40](#)

bond, [40](#)

## C

convert, [40](#)

cyc, [41](#)

## D

del, [41](#)

## E

exp, [41](#)

## F

fix, [42](#)

free, [42](#)

## G

gen, [42](#)

gline, [44](#)

## I

imple, [44](#)

## L

load, [44](#)

## P

prop, [45](#)

psconvert, [46](#)

## R

range, [47](#)

restore, [50](#)

## S

set, [50](#)

start, [51](#)

## V

vbox, [51](#)

vbox2jpg, [52](#)

## W

wall, [51](#)