

# VBOX 手册 v1.3

李长圣

2019 年 04 月 01 日

保护环境, 从阅读电子文档开始!

# 目 录

第 1 章 简介	7	第 5 章 构造模拟	19
第 2 章 Windows 下运行	9	第 6 章 命令参考	27
第 3 章 一个示例学会 VBOX	11	第 7 章 颜色表	39
第 4 章 提交计算	15	第 8 章 开通账户	41
4.1 直接提交 . . . . .	15	第 9 章 Linux 命令行	43
4.2 LSF 调度 . . . . .	15	第 10 章 致谢	45
4.3 SGE 调度 . . . . .	16		

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。

### **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 1.3 发布于 2019-01-10。

- 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？因为 VBOX 有如下特点：

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 下运行

1. 使用 [xshell](#) 连接安装了 VBOX 的服务器
  2. 使用 [xftp](#) 在服务器和本地电脑间传输数据
  3. 使用 [paraview](#) 查看 vtk 格式计算结果
  4. 使用 [gsview](#) 查看 ps 格式计算结果
- 使用 xshell 连接服务器
    - 文件-新建
    - 连接 *nju* 主机 *hpcc.nju.edu.cn* 端口号 *8801*
    - 连接-用户身份验证用户名: *zhang\_san* 密码 *123456*
    - 登陆成功后, 输入:

```
vboxdaily -v
```

输出:

```
VBOX 1.3(2018-10-28) (Education @ Nanjing University)
```

成功!

---

**注解:** 帐号申请办法见[开通账户](#)

---

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。

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  $4/3\pi r^3$  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
```

(下页继续)

(续上页)

```

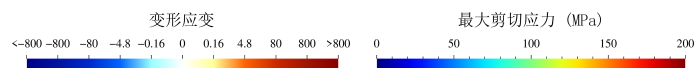
PROP DENSITY 2.5e3, fric 0.0, shear 2.9e9, poiss 0.2, damp 0.4, hertz
# 设置时间步及重力加速度
SET DT 5e-2, GRAVITY 0.0, -9.8
# 设置每 1000 步保存一次 ps 格式的计算结果
SET ps 1000
# 设置每 1000 步保存一次 dat 格式的计算结果
SET print 1000
# 沉积, 计算 5000 步
CYC 5000
# 删除 6000 米以上的颗粒
DEL RANGE y 4000.0 999000.0
# 平衡, 计算 1000 步
CYC 1000
# 输出包含颗粒的 [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 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 16000.0, 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
# 计算停止
STOP

```

计算结束后, 将得到以下结果:



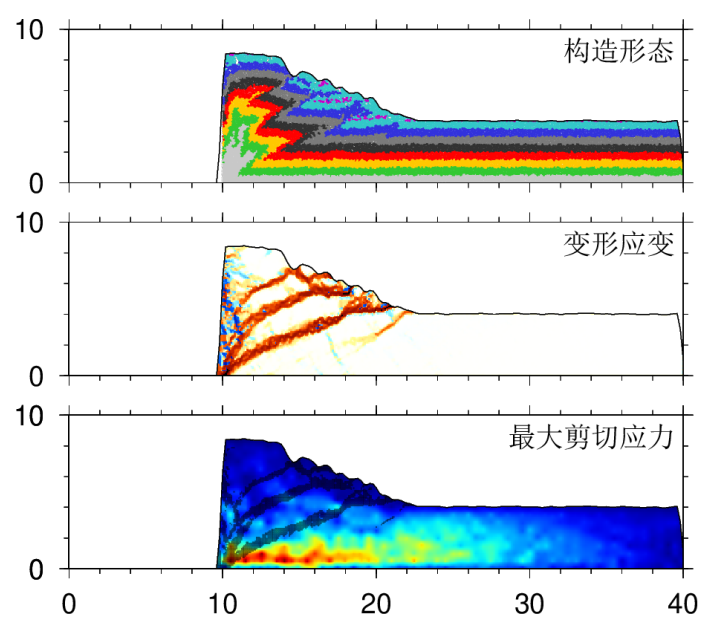


图 3.1: 单位 (km)

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 为你要运行的脚本名

### 4.1 直接提交

```
# 程序名 脚本名  
vboxdaily 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  
bjobs  
bkill <id>
```

- 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 -q mpi  
#BSUB -n 24  
#BSUB -R "span[ptile=24]"  
/share/home/hwyin/vbox/vboxdaily push.py
```

#### 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 1
↪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。

### 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集群安装了该调度系统

```
qsub sge.sh
qstat
qdel <id>
```

- 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
#2017-04-05
#LI ChangSheng @ NanJing Uninversity
#sheng0619@163.com
# 演示如何用 SGE 调度 VBOX
#$ -S /bin/bash
#$ -N example # 设置任务的名字
#$ -j y
#$ -M sheng0619@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 16 # 调用 16 个核计算
export OMP_NUM_THREADS=$NSLOTS
time vboxdaily cmd.py # 提交计算

```

## 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      16

```

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。

保护环境,从阅读电子文档开始!

## 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。

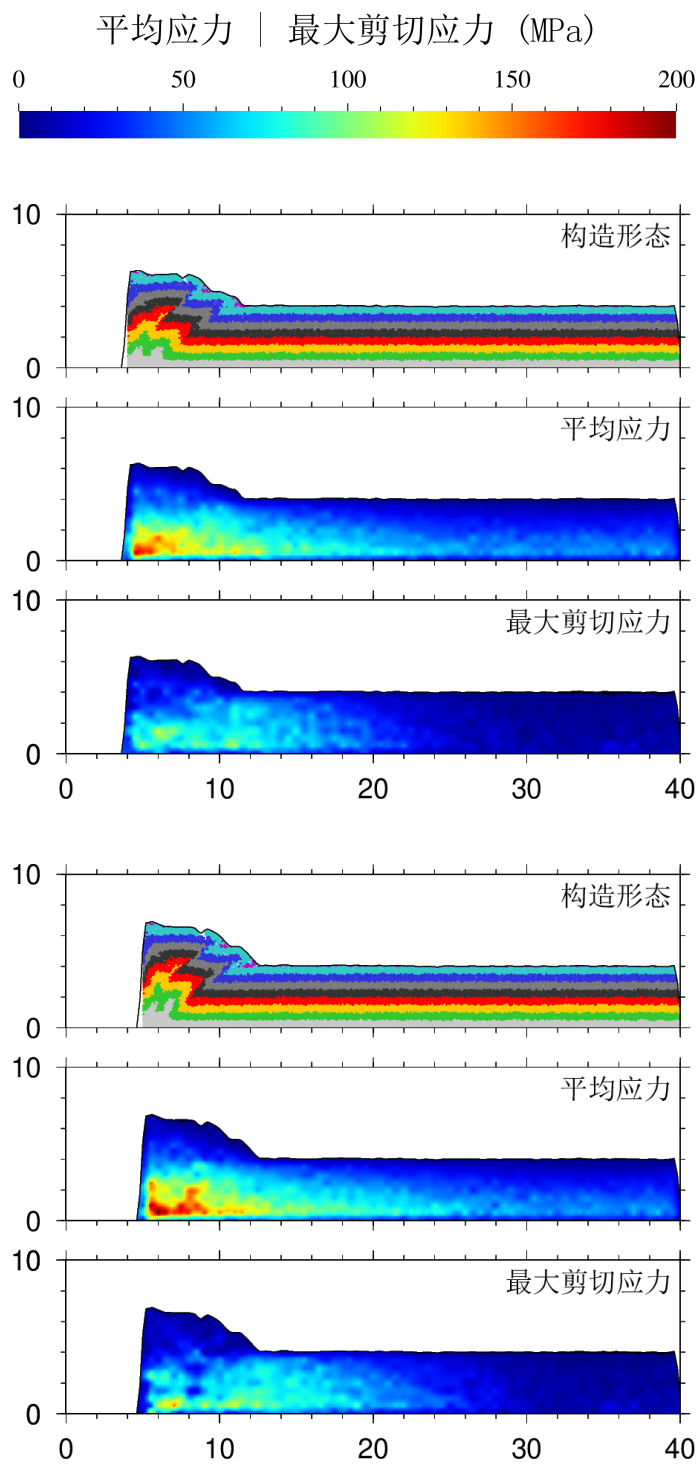
登陆南京大学高性能计算中心集群,运行 `syn_erosion.py` 脚本:

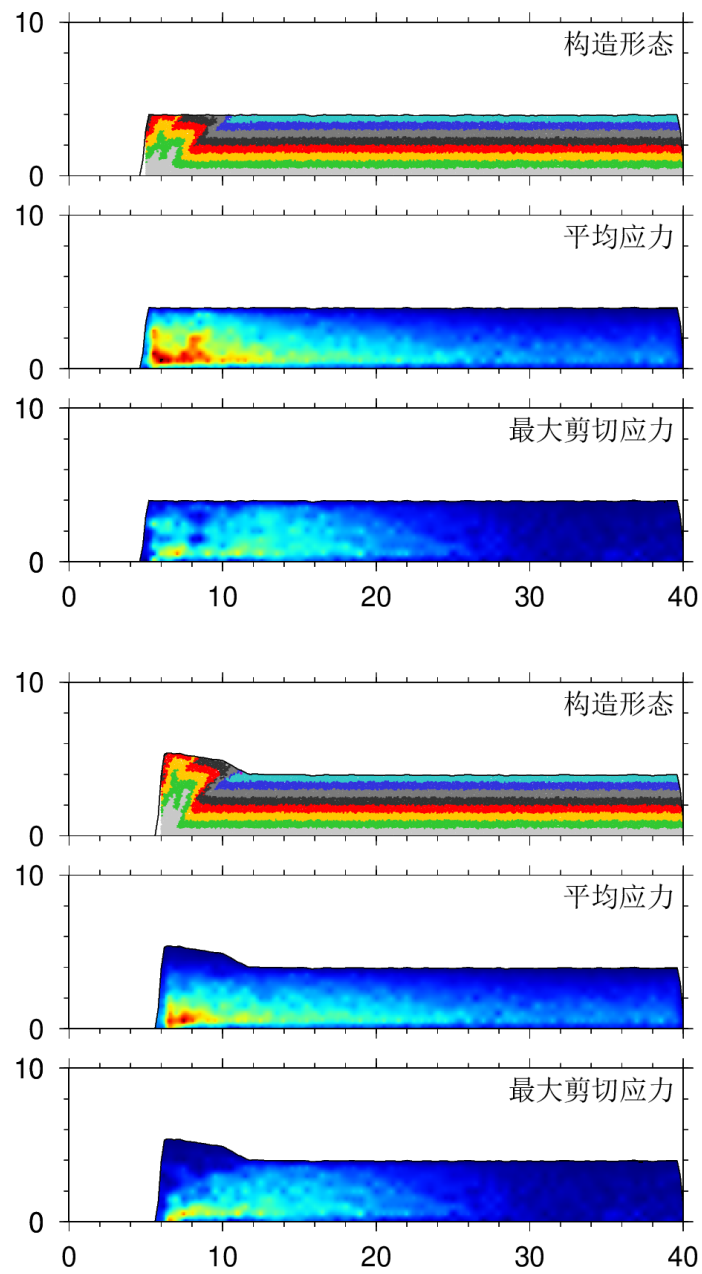
```
vboxdaily syn_erosion.py
```

`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:





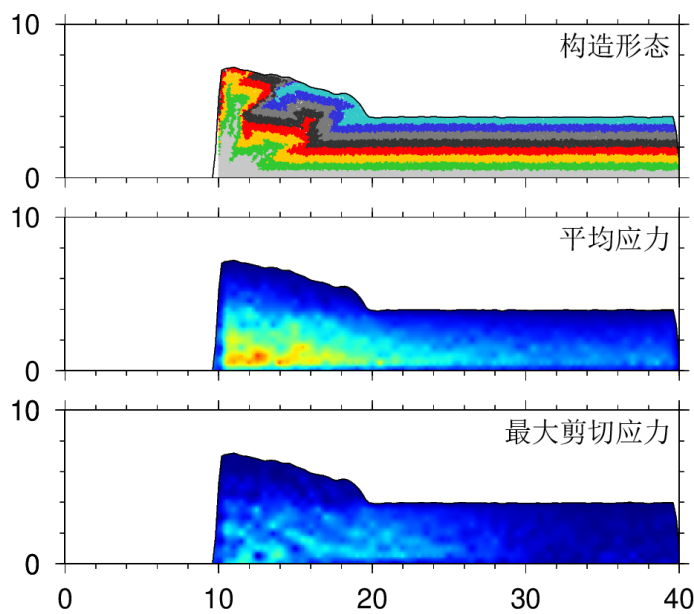


图 5.1: 单位 (km)

## 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:

```
vboxdaily 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)

```

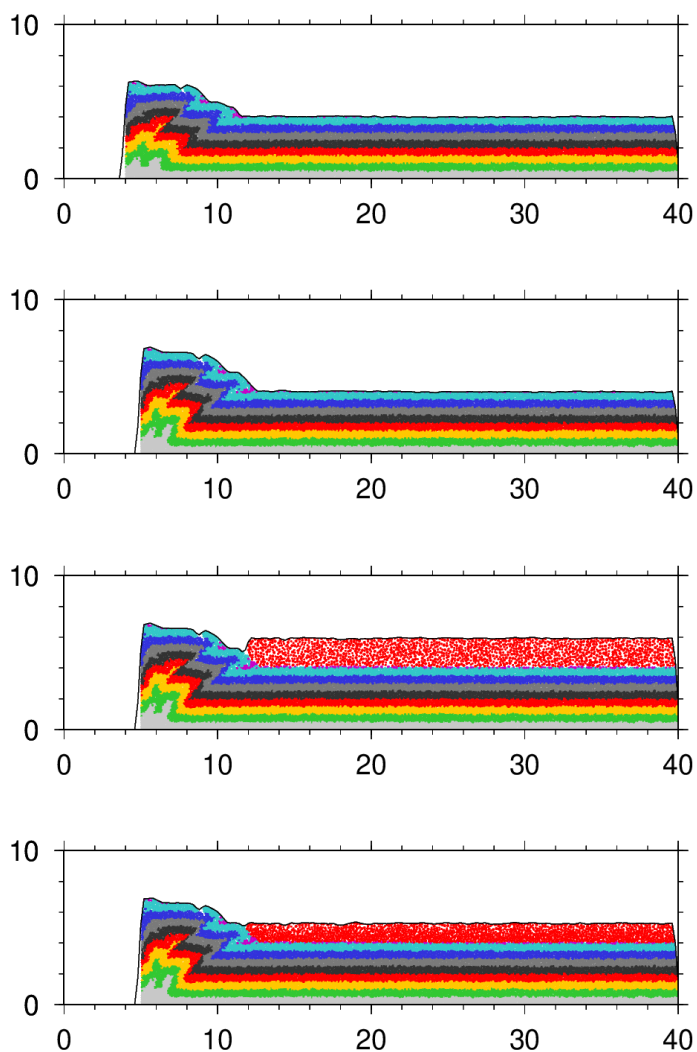
# 设置沉积颗粒 GROUP=sed 的微观参数
PROP DENSITY 2.5e3, fric 0.3, shear 2.9e9, poiss 0.2, damp 0.4, hertz range GROUP sed
# 计算 2000 步, 让颗粒沉积下来
SET print 100 # 每 100 步输出一次计算结果
CYC 2000
#####

# 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:



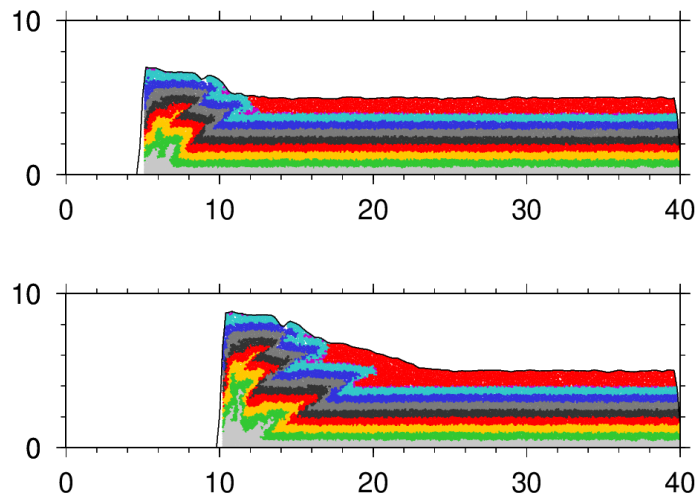


图 5.2: 单位 (km)

## 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:

```
vboxdaily 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_
↪0.0 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
```

(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:

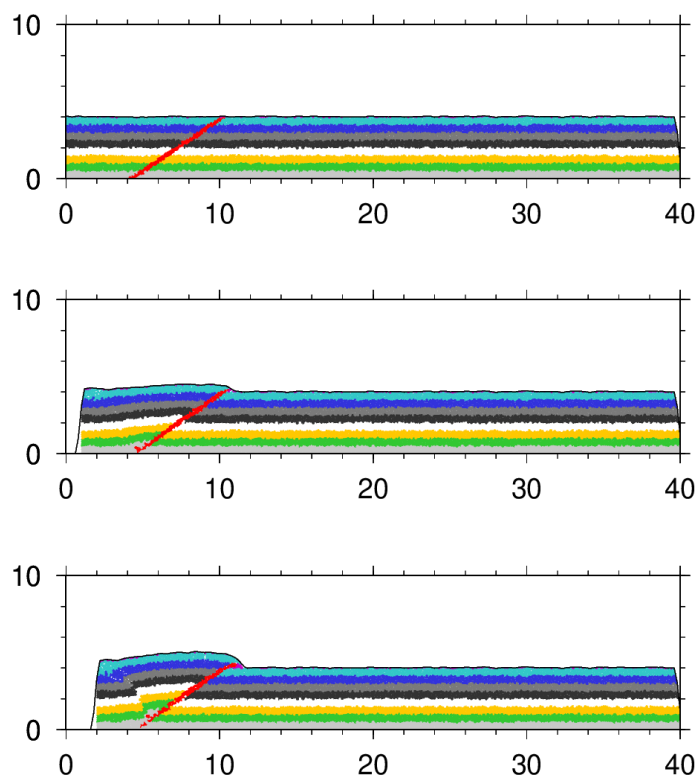


图 5.3: 单位 (km)

## 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
  - *vboxdaily*: `vboxdaily cmd.txt`, 运行 `cmd.txt` 脚本
- 格式转换
  - *vboxplot*: `vboxplot ./data`, 将 `./data` 中的计算结果绘制成 jpg 格式
  - *gmt psconvert*: 将 VBOX 生成的 PS 文件转换为其他图片格式。GMT 命令

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
- 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
- 范围圈定
  - *range*: 用在 prop bond del 后, 拥有 矩形 xy 多边形 P4 圈定方法
- 设置挤压量
  - *cyc* : 从网格文件中提取基本信息
  - *imple* : 从表数据中提取信息

## 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
```

## 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
```

## gen

**说明** gen 在一个矩形空间中生成一定数量的颗粒

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 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
```

## 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> NODES ( <float> <float> ) ( <float> <float> ), KN=<float> KS=<float>.
↪FRIC=<float>, [COLOR=<str>]
```

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
GLINE RAD=0.5 nodes ( 2.0 10.0 ) ( 40.0 10.0 ), kn=4.14e3 ks=4.14e3 fric=0.55.
↪color=red
```

## 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
```

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

## 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
```

实例:

```
# 设置材料参数 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, fric 0.0, shear 2.9e9, poiss 0.2, damp 0.0, hertz
# 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 range x 1.0 10.0 y 1.0 ↵
↵10.0
```

## psconvert

**简介** 将 GMT 生成的 PS/EPS 文件转换为其他图片格式

---

**注解:** 本模块原名为 `ps2raster`, v5.2.1 版本起更名为 `psconvert`

---

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[u][<margins>][-][+p<paint>][+p[<pen>]][+r][+s[m]|S<width>[<u>]/<height>**

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
```

-Au 会先将 GMT 的时间戳去掉再裁剪。

默认的裁剪方式会将图片裁剪到尽可能小。如果想要图片周围有额外的白色区域, 可以加上额外的边距 <margins>。其中 <margins> 可以取:

1. 一个数, 四条边的额外边距相同, 如 -A0.5c
2. 两个数字, 分别指定 X 和 Y 方向的额外边距, 如 -A0.5c/1c
3. 四个数字, 分别指定上下左右四条边的边距, 如 -A0.5c/0.5c/0.5c/0.5c

-A+s 可以指定最终图片的尺寸:

- -A+s<width> 指定最终生成的图片的宽度, 高度自动决定。程序会对图片做插值以保证 -E 的值
- -A+S<scale> 指定图片的缩放比例
- -A+sm<width>/<height> 设置图片所允许的最大尺寸。若原始图片的宽度不大于 <width> 则使用图片的原始尺寸, <height> 同理。

如下命令会生成一个宽为 5 厘米的图片:

```
gmt psconvert -A+s5c test.ps
```

- -A+r 使得在计算边界时使用 round 函数而不是 ceil 函数, 这会对裁剪造成极

其微小的区别, 仅当要处理非常小的图片时才需要使用。

- **-A+g<paint>** 为 BoundingBox 指定填充色
- **-A+p<pen>** 为 BoundingBox 指定边框颜色, 默认颜色为 0.25p,black
- **-A-** 覆盖 **-W** 选项中自动设置的 **-A** 值

**-C<gs\_options>** 传递给 GhostScript 的选项

该选项用于在调用 GhostScript 时传给 GhostScript 额外的选项, 若要额外给 GhostScript 增加多个选项, 可重复使用 **-C** 命令。

在 Windows 下, 若 PS 文件中含中文, 则可能需要使用 **-C** 选项告诉 ghostscript 字体路径:

```
gmt psconvert -C-sFONTPATH=C:\Windows\Fonts chinese.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

**-F<out\_name>** 指定输出的文件名

默认情况下, 该命令会直接使用输入的 PS 文件名, 并修改其后缀作为输出文件的文件名。比如 **test.ps** 转换出的图片则为 **test.jpg**、**test.png** 之类的。

**-F** 选项可强制指定输出文件名, 文件后缀由输出的文件格式自动决定:

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

**-G<ghost\_path>** 指定 GhostScript 可执行文件的路径

**psconvert** 底层调用了 GhostScript 来实现 PS 到其他格式的转换, 因而成功转换的前提是必须能够找到 GhostScript 的可执行文件。**-G** 选项即用于显式指定 ghostscript 可执行文件的路径。

说明:

1. Linux 下一般不需要设置 ghostscript 的路径, 除非你自己重新编译了 gs 并安装到了非标准路径下



2. Windows 下, 一般也不需要使用该选项, 程序会自动从注册表里获取路径信息
3. 如果从注册表中获取路径失败, 则必须指定 **-G** 选项, 例如

**-Gc:\programs\gs\gs9.02\bin\gswin64c**

**-I** Enforce gray-shades by using ICC profiles. GhostScript versions  $\geq 9.00$  change gray-shades by using ICC profiles. GhostScript 9.05 and above provide the `'-dUseFastColor=true'` option to prevent that and that is what **psconvert** does by default, unless option **-I** is set. Note that for GhostScript  $\geq 9.00$  and  $< 9.05$  the gray-shade shifting is applied to all but PDF format. We have no solution to offer other than upgrade GhostScript.

**-L<listfile>** `<listfile>` 中包含要转换的所有 PS 文件名

**-P** 强制转换后的图片为 Portrait 模式。

若 PS 文件在绘图时使用了 **-P** 选项, 则 **psconvert** 的 **-P** 选项无效; 若 PS 文件在绘图时未使用 **-P** 选项, 则 **psconvert** 的 **-P** 选项会将图片旋转 90 度。

**-Q[g|t][1|2|4]** Set the anti-aliasing options for **graphics** or **text**. Append the size of the subsample box (1, 2, or 4) [4]. Default is no anti-aliasing (same as *bits* = 1).

**-S** 在执行 `ghostscript` 命令后, 将具体的命令打印到标准错误流中, 且保留转换过程中的所有临时文件。该选项主要用于调试。

**-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
```

**-W** 见官方文档

**-Z** 转换完成后删除输入的 PS 文件。若转换失败, 输入的 PS 文件不会被删除。

## 其他

1. 转换为 PDF、PNG 时使用 DEFALTE 压缩; 转换为 TIFF 时使用 LZW 压缩;
2. 对于小 dpi 的图件而言, 可以使用 **-Qt4 -Qg4** 使得文字和线条看上去更加尖锐;
3. psconvert 还可以用于其他命令生成的 PS 文件;
4. 新版本的 Word 已经支持直接插入 PDF 格式的图片, 高清且文件小, 推荐使用;

## FAQ

1. 错误信息:

```
GMT PS format detected but file is not finalized. Maybe a -K in excess? No output_
↪created.
```

出现这种错误的最常见原因是最后一个绘图命令里多了一个 **-K**, 即本该只使用 **-O** 选项关闭 PS 文件的, 但却使用了 **-K -O** 选项。

## BUGS

1. Windows 下会自动为文件名加双引号, 导致无法找到要转换的 PS 文件 (v5.1.1)
2. 使用 **-A** 选项时, **-C** 选项无效, 导致无法在 Windows 转换含中文的 PS 文件 (v5.1.1)
3. 不能使用 **-F** 选项, 即不能生成多页 PDF (v5.1.2)
4. **-Z** 选项无效 (v5.2.1)

## 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

- 组名 GROUP <str>

### 1. 矩形实例

```
# 圈定一个矩形范围, 左右边界 (1.0 10.0), 上下边界 (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. 四边形实例

```
# 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)
```

## 使用 AND OR

- AND 并, range X (<float> <float>), Y (<float> <float>) AND GROUP <str>, 颗粒在 X()Y() 并且在 GROUP 中
- OR 或, range X (<float> <float>), Y (<float> <float>) AND GROUP <str>, 颗粒在 X()Y() 或者在 GROUP 中

### 1. AND 实例

```
# 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 实例

```
# 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>
```

设置进度条刷新闻隔:

```
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>
```

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 2.000
```

## vboxdaily

说明 VBOX 主程序

- 运行 **vboxdaily <par>** 其中 <par> 可取:
  - `cmd.txt` 运行脚本 `cmd.txt` 中的命令, `cmd.txt` 为 ASCII 码格式文件
  - `-v` 打印 VBOX 版本信息
  - `-V` 同上
  - `-version` 同上
  - `--version` 同上
- 实例 **vboxdaily -v**

VBOX 1.3(2018-10-28) (Education @ Nanjing University)

## vboxplot

说明 读取 VBOX 生成的 `all_*.dat` 文件绘制如 jpg pdf 等格式的图片

- 运行 **vboxplot <dir>** <dir> 为 `all_*.dat` 所在目录。
- 实例 **vboxplot ./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。

保护环境,从阅读电子文档开始!

## 第 7 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



保护环境,从阅读电子文档开始!



## 第 8 章 开通账户

VBOX 安装在南京大学高性能计算中心, 请联系南京大学高性能计算中心开通账户使用。

- 南京大学高性能计算中心官网: <http://hpcc.nju.edu.cn>
- 南京大学高性能计算中心开户申请办法: <http://hpcc.nju.edu.cn/index.php/single/applyaccount.html>

---

**注解:** 软件使用不收取费用, 集群管理费为南京大学高性能计算中心收取。申请到账户之后, 需要让管理员把自己加入 earth 组。南京大学地球科学与工程学院教职工默认分到这个分组。

---

保护环境,从阅读电子文档开始!

## 第 9 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 10 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.

# 索引

## B

ball, [28](#)

bond, [28](#)

## C

cyc, [28](#)

## D

del, [29](#)

## G

gen, [29](#)

gline, [29](#)

## I

imple, [29](#)

## L

load, [30](#)

## P

prop, [30](#)

psconvert, [30](#)

## R

range, [34](#)

restore, [35](#)

## S

set, [35](#)

start, [36](#)

## V

vboxdaily, [37](#)

vboxplot, [37](#)

## W

wall, [37](#)