**CFDEM 使用手册 (CFDME User Guide)**

**非前言：**

这将是一本非官方，非正式的CFDEM (<http://www.cfdem.com/>) 使用手册。 该手册的编写将由CFDEM QQ 论坛 （109526446）的成员自发、无偿地完成。 所以，该使用手册目前只用于内部交流而非商业用途。该手册的编写也许要历时一年，也许要历时五年。但是，群主相信众人拾柴火焰高，该手册一定会对中国科研人员和学生使用开源软件OpenFOAM, LIGGGHTS, 和 CFDEM 有极大的帮助。即使在手册编写过程中，相信它也会对刚刚进入到这个领域的新人有所帮助。

**为什么要写CFDEM使用手册？**

故事要从2012年开始说起。当时CFDEM的使用刚刚兴起，可以参照的文献和使用说明少之又少，只有官网的材料可以下载使用。当时群主就是在这样的环境下一步步摸索，成长的。历经的无助，痛苦和挣扎，相信刚刚开始接触这个开源软件的你也深有体会。从开始安装，到算例运行，再到自己的研究项目应用，以及开源软件的二次开发，每一步都是不停出错，不停调试，不停的学习。幸运的是，当时，CFDEM开发的人员活跃在官网论坛，解答了不少疑问，这无疑对我是一个极大的帮助，也是开源精神的体现。但是，对于国内的使用者来说，进入官网论坛需要翻墙。虽然我相信对于大多数搞数值模拟的人员，翻墙是一件很简单的事情，但依旧不是特别方便。另外网页版论坛的回复时效性不够好，时常自己的帖子就石沉大海了。所以，在这样的情况下，2015年1月30号群主建立了该QQ群，旨在帮助新入CFDEM模拟领域的朋友能有一个平稳的起步和过渡。 到2017年，两年的时间，现有群成员发展到112人，相信还有更多的人会找到组织，加入论坛。当然，QQ论坛也有其局限性，那就是所讨论的问题不能很好的归纳留档。后加入的朋友如果有类似或者相同的问题，无法参照以前的讨论。因此，编写一本CFDEM使用手册，系统地梳理软件的安装，使用，以及二次开发。同时，该手册将讨论基础的CFD，DEM，和CFDEM的理论和模型，便于研究人员更好地理解开源软件的运行和编译。在附录部分，也会整理归纳部分常见问题，便于参考。

**如何编写手册？**

该手册将采用开放式的编写方式，任何个人都可以对手册内容进行添加和修改。首先进行手册框架搭建，然后对内容进行填充和编辑。群主会定期更新手册，任何群成员都可以自愿参与讨论，编辑和修订。所有参与人员都可将自己姓名写在第二页的名单下。另外，在手册编写过程中，特有名词尽量同时使用中英文，以增加可读性。英文的使用在一定程度上可以锻炼英语科技论文的写作，为英文论文投稿打下一定的基础。在编写过程中，如有需求，群主也会联系并寻求CFDEM的开发人员， Christoph Kloss (*christoph.kloss@cfdem.com*), Christoph Goniva ([*christoph.goniva@cfdem.com*](mailto:christoph.goniva@cfdem.com)), 和Josef Kerbl ([*josef.kerbl@dcs-computing.com*](mailto:josef.kerbl@dcs-computing.com))的帮助，以促进该开源软件在国内的使用。

**如何联系群主？**

如有任何问题和想法，可以在群内部讨论，也可以单独联系群主：

张堃，华盛顿州立大学

QQ: 358911264

Email: [zhangkun-sdu@hotmail.com](mailto:zhangkun-sdu@hotmail.com)

**参与编写人员名单**

祁风雷，爱荷华州立大学，qifl2008@gmail.com

**手册目录（框架）持续更新中**

目录

[第一章CFDEM 安装 （Installation） 5](#_Toc476899866)

[第二章. OpenFOAM 6](#_Toc476899867)

[第三章. LIGGGHTS 7](#_Toc476899868)

[第四章. CFDEM 8](#_Toc476899869)

[第五章 Paraview 后处理 9](#_Toc476899870)

[第六章. 实例应用（APPLICATIONS） 9](#_Toc476899871)

[第七章. 重要参考文献（List of Key References） 10](#_Toc476899872)

[附录：论坛问答（Q&A） 10](#_Toc476899873)

# 第一章CFDEM 安装 （Installation）

CFDEM软件会不定期更新，现有最新版本CFDEM 3.6.0于2017年1月18日更新。其耦合OpenFOAM-3.X和LIGGGHTS-3.6.0. 更多的更新信息可以查询：

<http://www.cfdem.com/cfdemrcoupling-version-history>

虽然CFDEM可以在MacOS系统以及MS Windows系统下的运行，但是其官方开发以及维护都是基于Linux Fedora或者Ubuntu系统。因此，本手册的安装说明以及CFDEM使用都是基于Ubuntu系统。考虑到OpenFOAM与Ubuntu系统的兼容新，建议安装Ubuntu 14.04LTS (Long-Term-Support) 系统 (<http://releases.ubuntu.com/14.04/>)。

*Ubuntu的官网安装说明：*

<https://www.ubuntu.com/download/desktop/burn-a-dvd-on-ubuntu>

<https://www.ubuntu.com/download/desktop/create-a-usb-stick-on-ubuntu>

*OpenFOAM的官方安装说明，请参照：*

<https://openfoam.org/download/3-0-1-ubuntu/>

*LIGGGHTS和CFDEM的官方安装说明，请参照：*

<http://www.cfdem.com/installation-tutorial>

<http://www.cfdem.com/system/files/githubaccess_public.pdf>

**太多链接已经看晕了？不知道如何下手了？**不要紧，下面的教程就叫你一步步安装系统，安装OpenFOAM，安装LIGGGTHS，安装CFDEM，安装Paraview，以及算例运行。

**1.1 Ubuntu System 安装及常用命令**

1. 找一台能联网的计算机，打开链接：<http://releases.ubuntu.com/14.04/>

2. 根据你要装Ubuntu系统的电脑配置（64位？还是32位?），选择 64-bit (AMD 64) desktop image, 或者 32-bit PC (i386) desktop image, 下载Ubuntu系统安装程序。



3. 你可以选择使用刻录DVD作为Ubuntu系统安装盘  
(<https://www.ubuntu.com/download/desktop/burn-a-dvd-on-ubuntu>)，

或者使用USB作为系统安装盘:  
（<https://www.ubuntu.com/download/desktop/create-a-usb-stick-on-ubuntu>）。

3.1 插入至少有2GB空间的USB

3.2 打开dash (Ubuntu 系统下的一个应用)，并搜索Startup Disk Creator

3.3 选择Startup Disk Creator应用

3.4 点击“Other…”，选择下载的ISO安装文件，点击open

3.5 选择USB，点击“Make Starup Disk”

3.6 重启电脑，点击F9(不同电脑设置可能不同)选择Boot Device-USB, 准备安装！

4. 按照指示，安装Ubuntu 14.04 LTS.

Tip: 安装Ubuntu的时候，一定要给系统足够的空间，越大越好！

**表-Ubuntu常用命令**

|  |  |
| --- | --- |
| **命令** | **描述** |
| Ctrl+Alt+T | 打开Terminal |
| cd+空格+文件夹名称 | 进入文件夹；如果文件夹名称过长，可以尝试: cd+空格+文件夹前几个字母+Tab 组合 |
| cd .. | 返回上一级文件夹 |
| sudo | 使用超级用户权限运行命令 |
| ls | 显示该位置下的所有文件夹 |
| Ctrl + P | 运行上一步的命令 |

**1.2 OpenFOAM 安装** (<https://openfoam.org/>) (<https://openfoam.org/download/3-0-1-ubuntu/>)

(<https://openfoamwiki.net/index.php/Installation/Linux/OpenFOAM-3.0.x/Ubuntu#Ubuntu_14.04>)

参照OpenFOAM的安装说明，我们将使用terminal (Ctrl+Alt+T) 来进行安装New Installation. 使用黄色覆盖的语句需要复制并粘贴到terminal进行运行。

（1）复制，粘贴 下列语句到terminal：

sudo add-apt-repository <http://dl.openfoam.org/ubuntu>

sudo sh -c "wget -O - http://dl.openfoam.org/gpg.key | apt-key add -"

（2）更新apt

sudo apt-get update

（3）安装OpenFoam

sudo apt-get -y install openfoam30

（4）安装Paraview （后处理软件）

sudo apt-get -y install paraviewopenfoam44

**用户设置：**

（5）打开 .bashrc

gedit ~/.bashrc

（6）在bashrc文件最后添加下列语句，并保存

source /opt/openfoam30/etc/bashrc

（7）打开一个新的terminal (Ctrl+Alt+T), 复制粘贴下列语句到新的terminal

simpleFoam -help

（8）如果新的terminal出现Usage信息，说明你的安装和设置已经完成

（9）使用新的terminal，新建一个文件夹。

mkdir -p $FOAM\_RUN

你会发现你的$HOME文件夹下面多了一个OpenFOAM文件夹

（10）**算例运行。**一次将下列语句复制粘贴到terminal:

cd $FOAM\_RUN

cp -r $FOAM\_TUTORIALS/incompressible/simpleFoam/pitzDaily .

cd pitzDaily

blockMesh #划分网格

simpleFoam #运行simpleFoam Solver

paraFoam #打开paraview 进行后处理

进入到paraview, 点击apply, 生成模拟的模型和结果。可以查看速度u和压强p。paraview具体的教程可以参照chapter 5。

现在，如果你能成功运行该算例。恭喜你！已经安装好OpenFOAM了。

此时，你的OpenFOAM安装在opt文件下，通过terminal，使用cd / 进入 /$ 目录，使用 ls 显示所有的文件，cd opt/openfoam30 进入OpenFOAM文件夹。

如果实际进入OpenFOAM文件夹？ 点击Computer, 双击 opt，双击openfoam30。进入到openfoam30文件夹之后，application文件夹，src文件夹，和tutorials文件夹是以后经常进入的。

依据群成员王泽坤所编写的安装文档，使用terminal安装下列耦合必须的程序：

sudo apt-get -y install git-core build-essential flex bison cmake zlib1g-dev qt4-dev-tools libqt4-dev gnuplot libreadline-dev libncurses-dev libxt-dev libscotch-dev libopenmpi-dev libcgal-dev octave python-numpy firebird-dev libmysqlclient-dev libpq-dev libsqlite0-dev libsqlite3-dev git build-essential

**1.2.1 ThirdParty 3.0.x 设置**

cd OpenFOAM

**1.3 LIGGGHTS 安装  
(**<http://www.cfdem.com/installation-tutorial>)

(<https://www.cfdem.com/system/files/githubaccess_public_1.pdf>)

(1) 上述安装完成后，在terminal输入：

which mpirun

terminal应该显示/usr/bin/mpirun

(2) 在$HOME路径下新建LIGGGHTS文件夹：

cd #返回到$HOME路径下

mkdir LIGGGHTS #在$HOME路径下新建文件夹LIGGGHTS

cd LIGGGHTS #进入LIGGGHTS文件夹

(3) 下载并安装LIGGGHTS

git clone https://github.com/CFDEMproject/LIGGGHTS-PUBLIC.git

下载完成后, 进入LIGGGHTS/LIGGGHTS-PUBLIC/src 文件夹

cd LIGGGHTS-PUBLIC/src # 进入src文件夹

make clean-all # 清理原有编译

make fedora #进行编译

如果需要liggghts的二次开发，在编写完 XX.h 和 XX.cpp文件之后，可以使用上述清理和安装命令重新编译。

(4) 对LIGGGHTS建立symbolic link

cd #返回到$HOME路径下

sudo ln -s $HOME/LIGGGHTS/LIGGGHTS-PUBLIC/src/lmp\_fedora /usr/bin/liggghts

注意lmp\_fedora 和/usr 之间有个空格！！

which liggghts #应该显示/usr/bin/liggghts

恭喜你！LIGGGHTS已经安装成功了！可以喝杯咖啡，运行算例啦！

**1.3.1算例运行**

LIGGGHTS下载的程序里有一些自带的算例，是初学者很好的学习材料。可以通过研究算例，了解LIGGGHTS运行程序 in.XXX的编写。这里只是通过运行一个算例，来测试LIGGGHTS的安装情况。

进入算例：

cd #返回到$HOME路径下

cd LIGGGHTS/LIGGGHTS-PUBLIC/examples/LIGGGHTS/Tutorials\_public #进入算例文件夹，在这个步骤中，可以经常使用 ls 来显示目前目录下面所有的文件夹

cd packing #进入packing算例

在packing 算例文件夹下，有一个post文件夹，用来保存生成的结果。有一个in.packing程序，这是LIGGGHTS运行的文本程序，类似MATLAB中的m文件。有一个postscript和runscript，一个是后处理程序，一个是运行程序。在官网安装说明中，可以在terminal中输入liggghts < in.packing来运行算例。这里，群主喜欢使用./runscript来运行算例。原理都是一样的。来吧，在terminal中进入packing文件夹，输入：

./runscript #让你的程序奔跑吧！

现在LIGGGHTS可以运行了，但是需要安装后处理程序lpp将LIGGGHTS dump文件转化为VTK文件，才可以在paraview中查看结果。

1.3.2 LIGGGHTS后处理程序(post-processing)安装: lpp (LIGGGHTS Post-Processing)程序

(<https://www.cfdem.com/post-processing-liggghtsr-simulations>)

(1) git下载：

cd #返回到$HOME路径下

cd LIGGGHTS #进入LIGGGHTS文件夹

git clone git://cfdem.git.sourceforge.net/gitroot/cfdem/lpp mylpp

cd mylpp #进入lpp文件夹

cd #返回到$HOME路径下

gedit ~/.bashrc #打开./bashrc文件

(2) 电脑自动使用文本工具打开bashrc文件，复制粘贴下列语句到bashrc文件的最后

export LPP\_DIR=$HOME/LIGGGHTS/mylpp/src

export LPP\_NPROCS=4

export LPP\_CHUNKSIZE=1

alias lpp='python -i $LPP\_DIR/lpp.py --cpunum $LPP\_NPROCS --chunksize $LPP\_CHUNKSIZE'

注意！第一句我们需要将路径改为$HOME/LIGGGHTS/mylpp/src。官网显示的路径是export LPP\_DIR=$HOME/LIGGGHTS/LPP/src。在安装和使用过程中，有些需要灵活掌握。

**(3)保存.bashrc文件，关闭该文件，重新打开一个terminal （或者在该terminal下运行source $HOME/.bashrc）**

好啦！lpp程序安装成功了！现在去刚刚运行的LIGGGHTS算例中把dump文件转化成VTK文件吧！

(1) cd LIGGGHTS/LIGGGHTS-PUBLIC/examples/LIGGGHTS/Tutorials\_public/packing

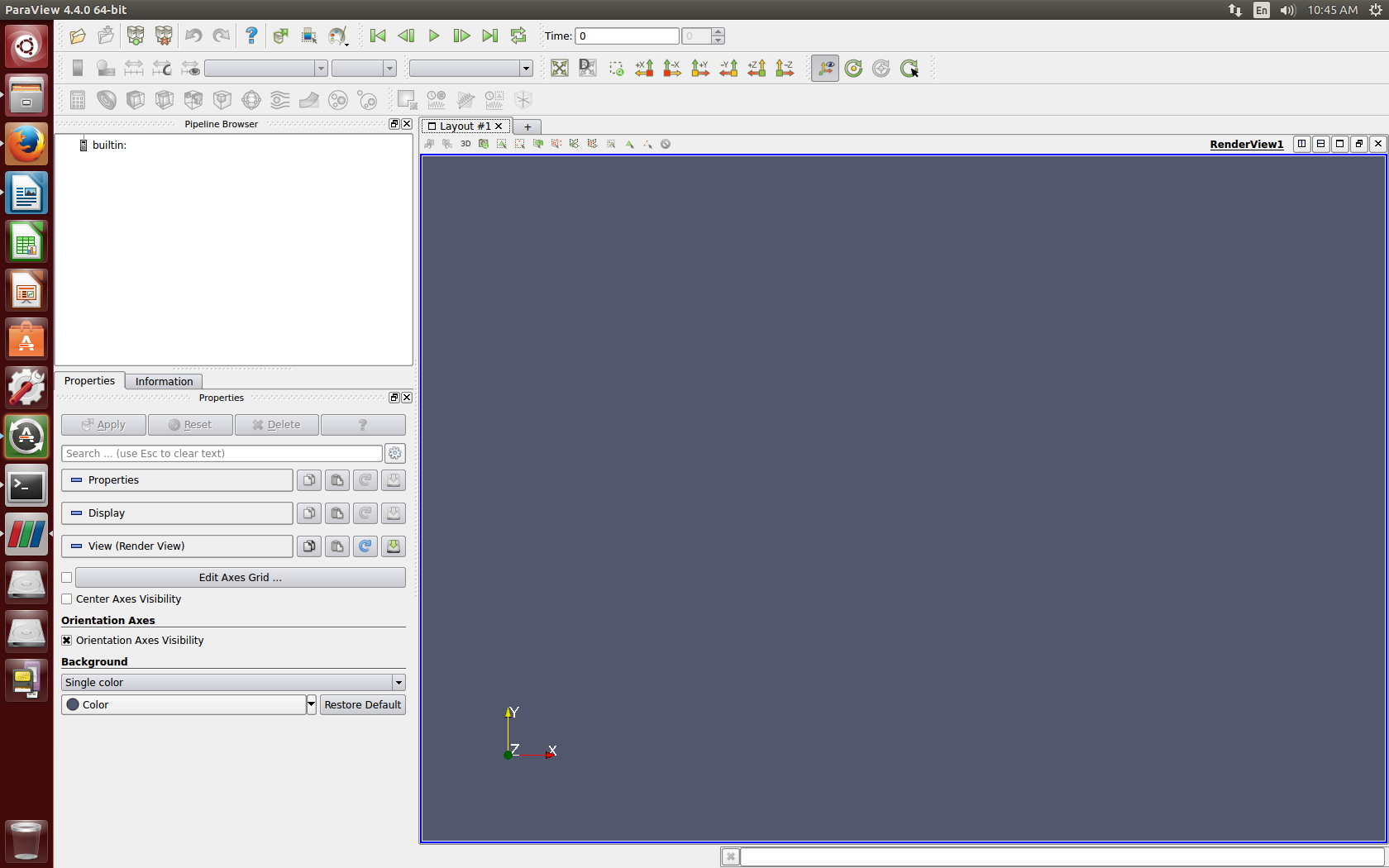
(2) cd post #刚刚运行的LIGGGHTS结果都保存在该文件夹下

(3) lpp dump\*.packing #把所有的dump文件转化为VTK文件

(4) Ctrl+D #等所有文件转化成VTK后，退出该运行

(5) paraview #我们在安装OpenFOAM的时候已经安装了paraview。如果你只需要使用LIGGGHTS，请参照前面的步骤安装paraview。

(6) 点击paraview左上角的open icon (文件夹形状)，按住Ctrl选择两个文件vtk文件，ok。Apply！



**此时，LIGGGHTS模拟的颗粒在paraview中显示的是“小点”。如何显示实际的颗粒大小？？？**

(1) 关闭原有的paraview并重新打开paraview。在paraview中选择Tools -> Manage Plugins

(2) 点击PointSprite\_Plugin 前面的“+”号展开

(3) 选中Auto Load，点击close关闭对话框

(4) 关闭paraview，重启paraview，选择 paraview左上角的open icon (文件夹形状)，按住Ctrl选择两个文件vtk文件，ok。Apply！

(5) 在Surface 下拉菜单中选择Point Sprite。此时颗粒变成大球球！但还不是实际尺寸！

(6) 在paraview左侧菜单栏，找到Scale by 在下拉菜单下找个radius。

WOO~错误提示了(可能你不会出现)：Shaders are not supported by this context! 不要惊慌，错误常有，使用GOOGLE! 在CFDEM官网群讨论中有解答

(<https://www.cfdem.com/forums/point-sprite-praview-visualizing-real-size-particles#comment-form>) 。(研究了半天，因为群主的workstations居然装的AMD的显卡)

另外，参照下面的论坛链接，群主安装了VTK-7.1.

(<https://www.cfdem.com/forums/liggghts-installation-guide-ubuntu-1204-lts-and-1304>)

1.3.3下载PIZZA （<http://www.sandia.gov/~sjplimp/download.html#pizza>）

cd

cd LIGGGHTS

mkdir PIZZA

手动将Downloads文件夹下的pizza.tar.gz文件剪切（cut），复制到新建的PIZZA文件夹下

cd PIZZA

tar –zxvf pizza.tar.gz #解压缩pizza文件，生成pizza-9Oct15

gedit ~/.bashrc

在bashrc文件后面添加

alias pizza=’python -i $HOME/LIGGGHTS/PIZZA/pizza-9Oct15/src/pizza.py’

保存，关闭

**1.4 CFDEM编译以及安装 (**<http://www.cfdem.com/system/files/githubaccess_public.pdf>)

现在，OpenFOAM和LIGGGHTS可以独立运行，但是需要CFDEM来进行耦合。

(1) 在$HOME路径下新建CFDEM文件夹

cd #返回到$HOME路径下

mkdir CFDEM

cd CFDEM

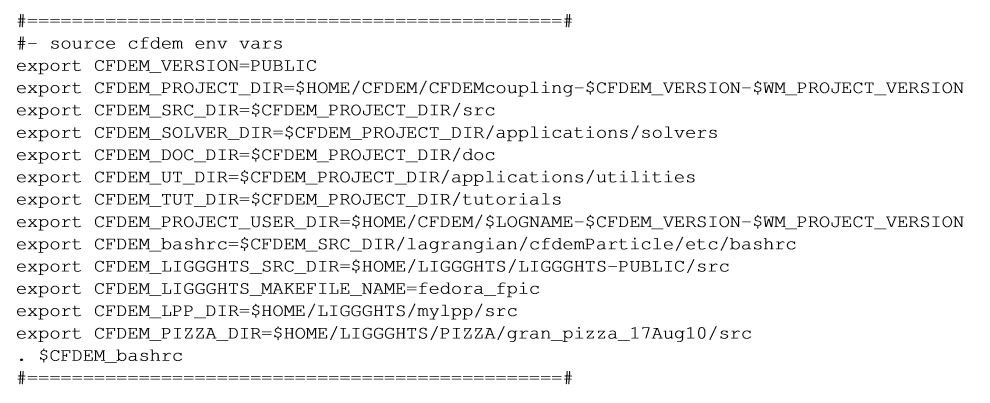
git clone git://github.com/CFDEMproject/CFDEMcoupling-PUBLIC.git CFDEMcoupling-PUBLIC-$WM\_PROJECT\_VERSION #下载CFDEM

(2) 设置环境变量

cd

gedit ~/.bashrc

将下列语句复制，粘贴到bashrc文件的最后 (参照源文件)



注意，前面我们的pizza的路径是$HOME/LIGGGHTS/PIZZA/pizza-9Oct15/scr。我们需要做相应修改！

另外，在export CFDEM\_VERSION=PUBLIC下一行添加

export WM\_PROJECT\_VERSION=3.0.1

保存bashrc文件，并关闭。

（3）打开新的terminal，弹出对话框是否要添加新的文件夹 PUBLIC-3.0.1 ? (y/n)

输入y, 点击回车。

进入  
CFDEM/CFDEMcoupling-PUBLIC-3.0.1/src/larangian/cfdemParticle/etc/addLibs\_universal, 修改additionalLibs\_3.0.x 为additionalLib\_3.0.1

（4）测试环境变量

cd

$CFDEM\_PROJECT\_DIR

$CFDEM\_SRC\_DIR

$CFDEM\_LIGGGHTS\_SRC\_DIR

当输入上述三个命令是，terminal应该显示“…is a directory”

(5) 测试CFDEM安装环境

cfdemSysTest

查看结果，所有“critical: yes”的路径，都必须是“valid:yes”。

(6) 耦合LIGGGHTS

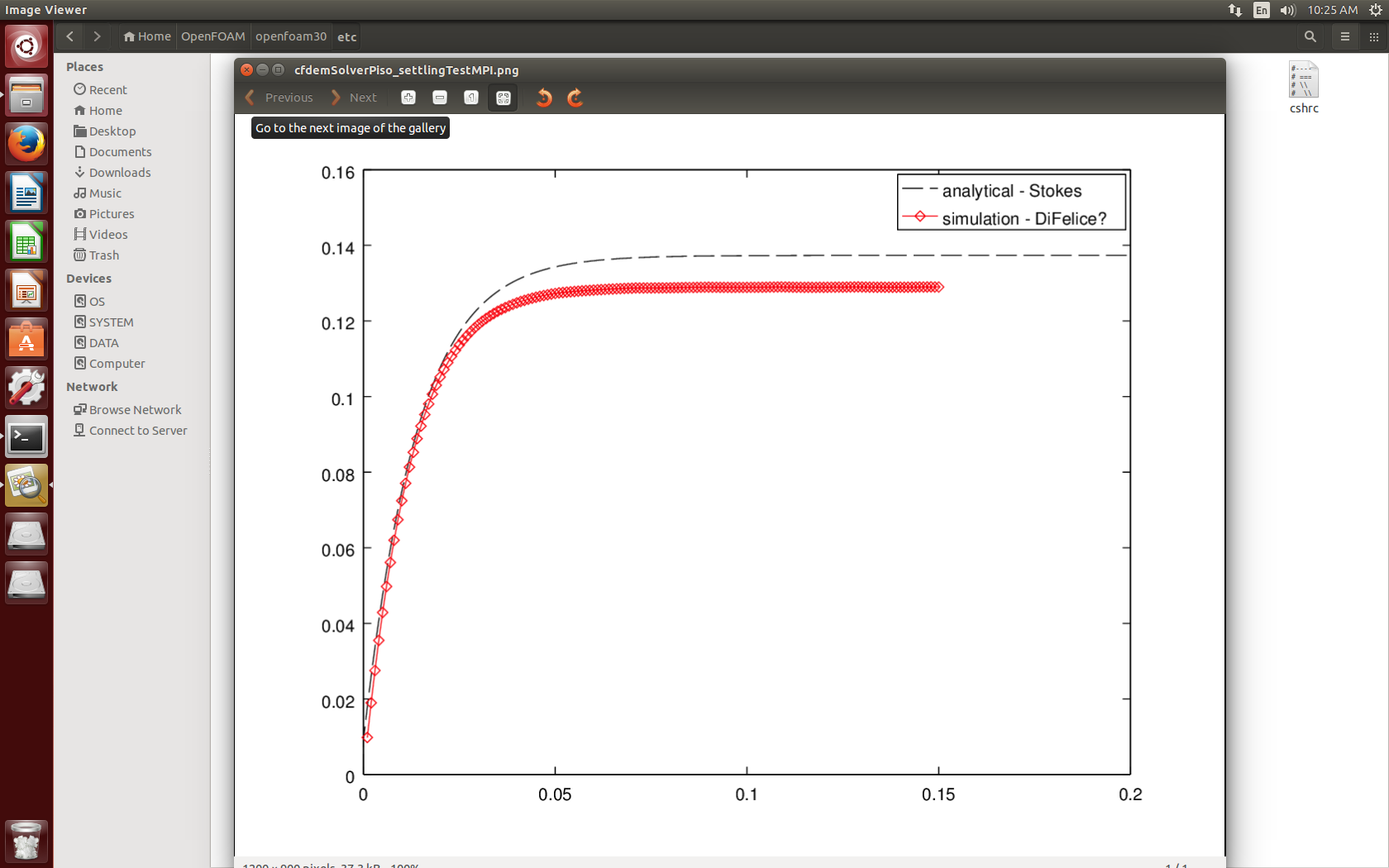
cfdemCompLIG

(7) 编译CFDEM

cfdemCompCFDEM

(9) CFDEM安装测试, 官方给出的测试命令为：

cfdemTestTUT



**如果看到了这个曲线，恭喜你，你的CFDEM已经安装成功了！**

在运行这个算例的时候，可能是你第一次运行CFDEM程序，可能会有这样那样的问题。不要紧，只要你安装过程中没出现错误，你的Solver都已经安装好就行了。下面简单介绍一下CFDEM的算例运行，用来进一步查看CFDEM是否已经安装成功。

**手动运行CFDEM算例**

cd CFDEM/CFDEMcoupling-PUBLIC-3.0.1/tutorials/cfdemSolverPisoScalar/packedBedTemp

ls #显示该算例文件夹内的程序，也可以手动进入该算例。在该算例中有CFD文件夹，有DEM文件夹，有Allrun.sh文件，有parCFDDEMrun.sh文件，有parDEMrun.sh文件。这些都是标准设置，可以进入参考，后面会有详细介绍。这里，进入parCFDDEMrun.sh文件，将cleanup=”true” 改为 cleanup=”false” 将runOctave=”true” 改为runOctave=”false”并保存。修改的目的是保存程序运行过程中的文件，可以后续查看。回到terminal，输入：

./Allrun.sh #根据该算例设置，先运行in.liggghts\_init文件 150000步，然后在耦合运算 1.5s (参考该算例CFD/system/controlDict文件)。如果算例可以正常运行，说明你的pisoScalar Solver 也正确安装啦！

cd DEM/post #进入该算例的DEM/post文件夹

lpp dump\*.liggghts\_run #转化dump文件为vtk文件

Ctrl+D

paraview

**1.5 Gmsh 安装 （网格划分软件）**

最简单的方法：进到Ubuntu Software Center, 搜索gmsh. 找到Gmsh Mesh Generator, 点击Install。

# 第二章. OpenFOAM

其实这一章应该放在后面，因为如果你只用OpenFOAM，估计不会对这一章节感兴趣。如果你只是想使用LIGGGHTS做颗粒模拟，可以直接跳到第三章。那为什么要把这一章节放在LIGGGHTS和CFDEM前面呢，因为理解OpenFOAM的源文件对使用CFDEM会有极大的帮助！

**2.1 CFD 基础理论**

The Computational Fluid Dynamics (CFD) 计算流体力学是一门预测流体流动（fluid flow），热传导（heat transfer）, 质量传导（mass transfer）, 化学反应 （chemical reactions）的计算科学。通过数值计算来求解用于描述上述过程的数学方程。 以流体为例：Based on the conservation equations, the accumulation of any quantity () in the control volume () over time step (Δt) = Net influx of into control volume – Net efflux of out of control volume + Net generation of inside control volume.

The “General Scalar Transport Equation” can be derived as:

In vector form:

where, is the storage term; is the convection term; is the diffusion term; and S is the source term (generation per unit volume W/m3).

where, is a specific or general quantity; is density; is velocity vector; and is the diffusion coefficient.

当 时，上述方程变为 “Continuity Equation”:

当 时，上述方程变为“Momentum Equation”:

*结合群主的一个研究项目，关于水分在土壤中的运移，来推到上述偏微分方程的数值求解，并结合该实例描述OpenFOAM的二次开发。*

当 时， 上述方程变为：

For one-dimensional case,

where, ; If put

求解上述偏微分方程， CFD一般使用有限体积法（finite volume method）。For the equation above, integrate over time () and volume ()

where, is the RH at next time; and is the RH at current time.

where, ; and use which is the C at current time by implicit method.

For 1-D case, and ; and

Because here we use implicit method,

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  | W (i-1) | P (i) | E (i+1) |  |  |  |  | N |

E and W are the East cell and West cell of the P Cell (Present Cell).

Since , So,

Put ; ; ; and

上述公式变为

我们可以用Tridiagonal Matrix Algorithm (TDMA) 来求解上述方程：

设定P Cell 编号为i, then W cell 为i-1, E cell 为i+1; ; ; ; ;

For the boundary condition (边界条件), 对于该问题，假设第一个Cell的left boundary 的RH等于空气的RH=60% （第一类边界条件，Dirichlet Condition ()）. And initial condition (初始条件) at time=0, RH=99.99% for all the cells.

因此，对于第一个Cell,

对于最后一个Cell (N), 因为右边界封闭，因此, (第二类边界条件 Neumann Conditions).

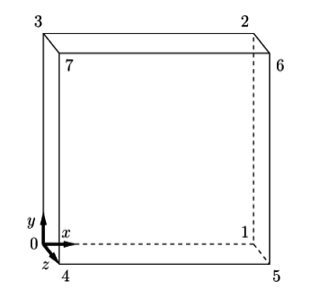
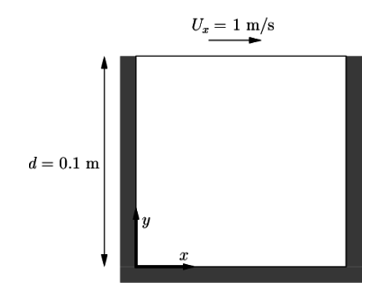
根据上述的推到，可以使用MATLAB进行编程。MATLAB的m文件附在附录\*。

**对于二维的运移情况（2D Case）, 偏微分方程（partial differential equation）变成：**

**2.2 OpenFOAM算例讨论** (<https://cfd.direct/openfoam/user-guide/>)

**2.2.1 OpenFOAM用户手册的经典算例（OpenFOAM-tutorials-incompressible-icoFoam-cavity）**

该算例使用icoFoam Solver (求解器)模拟lid-driven cavity, 如下图。该icoFoam Solver用于计算Transient，incompressible, laminar flow of Newtonian fluids。整个算例分为三个部分，前处理，运行，和后处理三个部分。



进入该算例，cavity里面有三个文件夹，分别是0，constant，和system。0文件夹里面的文件主要进行初始条件（压强p和速度U）的设置。constant文件夹内有transportProperties文件用于定义流体nu。system文件夹下有blockMeshDict用于定义流体域（fluid domain）和网格划分，controlDict用于定义要使用的solver以及时间变量。fvSchemes用于。fvSolution用于。OpenFOAM算例运行过程如下：

**(1)使用blockMesh命令生成网格。**

在OpenFOAM中，可以使用blockMesh定义网格属性。打开blockMeshDict文件：

/\*–––––––––––––––––––––––––––––––\*- C++ -\*–––––––––––––––––––––––––––––––––-\*\   
2  | =========                 |                                                 |   
3  | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |   
4  |  \\    /   O peration     | Version:  3.0.1                                  |   
5  |   \\  /    A nd           | Web:      www.OpenFOAM.org                      |   
6  |    \\/     M anipulation  |                                                 |   
7  \\*–––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––\*/   
8  FoamFile   
9  {   
10      version     2.0;   
11      format      ascii;   
12      class       dictionary;   
13      object      blockMeshDict;   
14  }   
15  // \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //   
16     
17  convertToMeters 0.1; //将1m转化成0.1m；下面定义的高度为1m，实际上是0.1m;  
18     
19  vertices //定义定点坐标  
20  (   
21      (0 0 0) //point 0  
22      (1 0 0) //point 1  
23      (1 1 0) //point 2  
24      (0 1 0) //point 3  
25      (0 0 0.1) //point 4  
26      (1 0 0.1) //point 5  
27      (1 1 0.1) //point 6  
28      (0 1 0.1) //point 7  
29  );   
30     
31  blocks //定义block  
32  (   
33      hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1) //(20 20 1) 定义了沿x，y，z方向的网格数量；simpleGrading定义网格沿各个方向轴扩展比例（cell expansion ratio），可以用于定义非均匀网格（graded or refined）；  
34  );   
35     
36  edges   
37  (   
38  );   
39     
40  boundary //定义边界  
41  (   
42      movingWall //边界名称  
43      {   
44          type wall; //边界类型 solid wall  
45          faces   
46          (   
47              (3 7 6 2) //所生成的长方体上面边界  
48          );   
49      }   
50      fixedWalls   
51      {   
52          type wall;   
53          faces   
54          (   
55              (0 4 7 3) //长方体左面  
56              (2 6 5 1) // 长方体右面  
57              (1 5 4 0) //长方体下面  
58          );   
59      }   
60      frontAndBack   
61      {   
62          type empty; //empty patch用于2D或者1D的模拟  
63          faces   
64          (   
65              (0 3 2 1) //长方体后面  
66              (4 5 6 7) //长方体前面  
67          );   
68      }   
69  );   
70     
71  mergePatchPairs   
72  (   
73  );   
74     
75  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

打开terminal，cd命令进入cavity算例。输入blockMesh，即可生成网格。使用paraFoam命令，打开paraview，点击Apply。surface下拉菜单选择Surface with Edges查看网格生成情况。

**(2)定义边界和初始条件**

在0文件夹里面，有p 文件(压强)和U文件（速度）用于定义初始和边界条件。例如打开p文件，

17  dimensions      [0 2 -2 0 0 0 0]; //定义变量的单位；OpenFOAM通过7个标量来定义变量的单位；在[]中，这七个标量代表[Mass(kg/lbm), Length(m/ft), Time(s), Temperature(Kelvin/Degree Rankine), Quantity (mole), Current (Ampere), Luminous intensity(candela)]. 该例子中，第二位为2，第三位为-2，代表m2s-2, 该压强为kinematic pressure (P=p/ρ0, 压强/流体密度)。   
18     
19  internalField   uniform 0; //因为压强为标量Scalar，所以只需一个数字定义内部流域  
20     
21  boundaryField //边界条件  
22  {   
23      movingWall //对应上面blockMesh中的上边界  
24      {   
25          type            zeroGradient; //normal gradient of P is zero (); Neumann Boundary Condition.  
26      }   
27     
28      fixedWalls //对应上面blockMesh中的左，右，和下边界  
29      {   
30          type            zeroGradient;   
31      }   
32     
33      frontAndBack //对应上面blockMesh中的前，后边界  
34      {   
35          type            empty; //empty边界用于2-D模拟  
36      }   
37  }   
38     
39  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

再看U文件（速度）：

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

dimensions [0 1 -1 0 0 0 0]; //Units = ms-1

internalField uniform (0 0 0); //因为速度是向量vector，所以使用(0 0 0) 来定义内部流域

boundaryField

{

movingWall

{

type fixedValue; //上边界流速固定 (fixedValue); Dirichlet Condition;

value uniform (1 0 0); //沿x方向为1 m/s;

}

fixedWalls

{

type fixedValue; //左，右，下边界流速固定(fixedValue)

value uniform (0 0 0); //沿各个方向的流速为0;

}

frontAndBack

{

type empty;

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

**(3)定义流体的物理性质physical properties**

**流体的运动粘度（kinematic viscosity）的定义在Cavity/constant/transportProperties文件中定义：**

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

nu [0 2 -1 0 0 0 0] 0.01; //nu (kinematic viscosity); Units = m2s-1; value=0.01;

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

该算例中，Reynolds number (Re)等于, d is characteristic length=0.1m, U=1m/s, nu=0.01 m2s-1. 所以Re=10, laminar flow, 可以用icoFoam 求解。

**(4) System 文件夹以及运行控制（control）设定**

在system文件夹中，除了讨论过的blockMeshDict文件, 还有controlDict, fvSchemes, 和fvSolution文件。先看controlDict文件：

18  application     icoFoam; //solver used  
19     
20  startFrom       startTime; //模拟从startTime开始；startTime=0s (下一句)；  
21     
22  startTime       0;   
23     
24  stopAt          endTime; //模拟结束时间；endTime=0.5s (下一句)；  
25     
26  endTime         0.5;   
27     
28  deltaT          0.005; //定义时间步长，0.005s  
29     
30  writeControl    timeStep; // write data every WriteInterval time steps  
31     
32  writeInterval   20; //每20个timestep, output one result; this case, 每0.1s (20\*0.005s) 写一个数据  
33     
34  purgeWrite      0; // purgeWrite 用于覆盖数据以节省空间；default=0， disable the purging。  
35     
36  writeFormat     ascii; //default  
37     
38  writePrecision  6; //6 is default value  
39     
40  writeCompression off; //specify whether files are compressed with gzip when written;  
41     
42  timeFormat      general;   
43     
44  timePrecision   6; //interger used in conjunction with timeFormat; 6 is default value  
45     
46  runTimeModifiable true; //allow the user to modify parameters during a simulation  
47     
48     
49  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

接下来看一下fvSchemes文件：

Recall what we have discussed before:

In vector form:

当 时，上述方程变为 “Continuity Equation”:

当 时，上述方程变为“Momentum Equation”:

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

ddtSchemes // first time derivatives (); d2dt2Schemes 适用于second time derivatives ()

{

default Euler; //Euler算法适用于transient，first order implicit, bounded;

}

gradSchemes //gradient

{

default Gauss linear; //这一类(gradient)的求解默认是Gauss Linear;

grad(p) Gauss linear; //for , Gauss specifies the standard finite volume discretisation of Gaussian integration which requires the interpolation of values from cell centres to face centres; linear means linear interpolation or central differencing.

}

divSchemes //divergence e.g. ; advection term;

{

default none; //这一类(divergence)的求解没有(none)默认的scheme。这一类中的每项都需要定义算法。

div(phi,U) Gauss linear; //phi generally denotes the volumetric flux of velocity on the cell faces for constant-density flows and the mass flux for compressive flows; div (phi, U) is the advection of velocity; linear for second order, unbounded;

}

laplacianSchemes //Laplacian or ; e.g ; diffusion term;

{

default Gauss linear orthogonal; //only linear interpolation is used for diffusion term; orthogonal is for regular mesh;

}

interpolationSchemes // cell to face interpolations of values

{

default linear; //primarily used in the interpolation of velocity to face centres for the calculation of flux phi;

}

snGradSchemes //Surface normal gradient schemes; component of gradient normal to a cell face;

{

default orthogonal; // orthogonal scheme requires a regular mesh; other schemes including corrected scheme (second-order accuracy, explicit non-orthogonal correction), limited scheme (0.33 or 0.5), and uncorrected scheme.

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

最后一个文件fvSolution:

18  solvers //specify each linear solver for each discretised equation; the term ‘linear-solver’ is abbreviated to ‘solver’ here.   
19  {   
20      p //for pressure  
21      {   
22          solver          PCG; //PCG for symmetric matrices; PBiCG for asymmetric matrices;  
23          preconditioner  DIC; // DIC is short for Diagonal incomplete-Cholesky (symmetric) and incomplete-LU (asymmetric); other options include FDIC, diagonal, GAMG, none.   
24          tolerance       1e-06; //the smaller residue is , the more accurate the solution; the residual falls below the solver tolerance  
25          relTol          0; // relative tolerance; the ratio of current to initial residuals falls below the solver relative tolerance; Relative tolerance is set as ‘0’ to force the solution to converge to the solver tolerance in each time step for transient simulations;   
26      }   
27     
28      U   
29      {   
30          solver          smoothSolver; //solver that uses a smoother  
31          smoother        symGaussSeidel; // required for smoothSolver; symGaussSeidel stands for symmetric Gauss-Seidel;   
32          tolerance       1e-05; //ditto  
33          relTol          0; //ditto  
34      }   
35  }   
36     
37  PISO // Pressure-implicit split-operator; a pressure equation is solved nCorrectiors within one time step according; Semi-implicit method for pressure-linked equations (SIMPLE); a combined PIMPLE. PISO and PIMPLE are used for transient problems and SIMPLE for steady-state.   
38  {   
39      nCorrectors     2; //do 2 corrections in each steps  
40      nNonOrthogonalCorrectors 0; //0 for steady-state  
41      pRefCell        0; //In a close incompressive system, pressure is relative. Set a pressure reference level in cell;  
42      pRefValue       0; //set a pressure reference value;   
43  }   
44     
45     
46  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

**(5) 运行算例**

在terminal中进入路径：openfoam30/tutorials/incompressible/icoFoam/caity。terminal中输入：

icoFoam //solver name; see ‘controlDict’ file;

(6) **查看结果(Post-Processing)**

terminal中输入：

paraFoam //打开paraview

点击Apply, 下拉菜单可以选择U 或者 p来查看速度或者压强结果。点击中间动态播放算例。

另外，可以使用foamToVTK 命令将OpenFOAM结果文件转化为VTK格式，再使用paraview进行查看。Paraview中可以使用Glyph来画速度矢量图，还可以显示速度或者压强沿某一直线的结果。这些可以自行摸索。该手册第五章提供了详细的介绍。

**2.2.2 Turbulent Flow Model**

OpenFOAM可以使用laminar模拟非湍流模型(no turbulence model), 还可以使用RAS (Reynolds-Averaged Stress) 模型和LES (Large-Eddy Simulation)模型模拟湍流的情况。对于RAS湍流模型，针对非压缩流体，OpenFOAM提供了kEpsilon（k-ɛ）模型，kOmega（k-Ω）模型，kOmegaSST模型，RNGkEpsilon模型等等。 结合tutorials/incompressible/pisoFoam/ras/cavity 算例，看看湍流模型的设置。该算例使用k-ɛ模型模拟Reynolds number ()等于104的情况 (d=0.1, U=1, nu=1e-5)。

**(1) blockMesh命令生成网格**

与上述算例相同，打开terminal：

cd OpenFOAM/openfoam30/ tutorials/incompressible/pisoFoam/ras/cavity

blockMesh

**(2) 定义边界和初始条件**

进入0文件夹，除了p和U文件之外，还有epsilon, k, nut, nuTida, omega文件。

**打开k文件**：

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

dimensions [0 2 -2 0 0 0 0]; //unit m2s-2

internalField uniform 0.00375; // k=, is the estimated fluctuating component of velocity.在正交坐标中，,该算例中假设.所以k=.算例应改成0.00375而非0.00325.

boundaryField

{

movingWall

{

type kqRWallFunction;//turbulent kinetic energy (k, q, R)的通用边界条件

value uniform 0.00375;

}

fixedWalls

{

type kqRWallFunction;

value uniform 0.00375;

}

frontAndBack

{

type empty;

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

**打开ɛ文件**：

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

dimensions [0 2 -3 0 0 0 0]; //Unit m2s-3

internalField uniform 0.00754;//;是k-ɛ的常数=0.09；k=0.00375;*l*是turbulent length scale=5%\*0.1=0.005；所以ɛ=7.54 X 10-3 m2s-3.

boundaryField

{

movingWall

{

type epsilonWallFunction;//boundary condition for epsilon

value uniform 0.00754;

}

fixedWalls

{

type epsilonWallFunction;

value uniform 0.00754;

}

frontAndBack

{

type empty;

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

**nut 文件(turbulent viscosity field ())：**

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

dimensions [0 2 -1 0 0 0 0];

internalField uniform 0;

boundaryField

{

movingWall

{

type nutkWallFunction;//其他的wall function models 还有nutRoughWallFunction；

value uniform 0;

}

fixedWalls

{

type nutkWallFunction;

value uniform 0;

}

frontAndBack

{

type empty;

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

其他的文件Omega用于k-Omega模型，nuTilda用于Spalart-Allmaras模型，这两个文件在该算例中属于多余的文件，可以删除！

**(3)定义流体的物理性质以及湍流模型**

在constant文件夹下面，transportProperties文件与前述算例相同，主要用于定义运动粘度（kinematic viscosity），该湍流算例中，运动粘度为1e-05 m2/s.所以Re=104

打开turbulenceProperties文件

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

simulationType RAS;//使用RAS模型；这里可以改成laminar用来算非湍流模型,或者LES模型

RAS

{

RASModel kEpsilon; //选择RAS模型中的湍流模型，结合算例这里应改为kEpsilon；而非使用kOmega;

turbulence on;//算湍流时，设置为on；

printCoeffs on;//对于每个湍流模型的系数，它们保存在相应的文件中并有默认值。设置printCoeffs为on确保这些默认值会正常输出。在OpenFOAM中这些默认值也可以根据自己的情况进行调整。

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

**(4) System 文件夹以及运行控制（control）设定**

这里主要讨论一下controlDict, 其他文件fvSchemes和fvSolution可以结合前面的分析自行研究。

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

application pisoFoam; //solver used, pisoFoam

startFrom startTime;

startTime 0;

stopAt endTime;

endTime 10;

deltaT 0.005;//设置确保Courant Number ;;, cell size in the direction of velocity, 20是mesh number；所以；

writeControl timeStep;

writeInterval 100;

purgeWrite 0;

writeFormat ascii;

writePrecision 6;

writeCompression off;

timeFormat general;

timePrecision 6;

runTimeModifiable true;

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

**(5) 运行算例**

terminal进入该算例路径，输入：

pisoFoam

(6) **查看结果(Post-Processing):** 同样，使用paraFoam查看结果，或者使用foamToVTK将结果转化成VTK格式，然后用paraview查看。

**2.2.3 Channel Flow in Cylinder (Block Mesh)**

**2.3.4 Impinging Jet Flow Simulation (k-Omega模型)**

**2.3 OpenFOAM 二次开发**

Thermal temperature example

# 第三章. LIGGGHTS

**3.1 LIGGGHTS基础理论及模型**

**3.2 LIGGGHTS算例讨论**

**3.3 LIGGGTHS二次开发**

# 第四章. CFDEM

**4.1 CFDEM 基础理论及模型**

**4.2 CFDEM算例讨论**

**4.3 CFDEM二次开发**

# 第五章 Paraview 后处理

5.1 使用Paraview提取数据 (data)

5.2 使用Paraview制作动画 (animation)

# 第六章. 实例应用（APPLICATIONS）

6.1

# 第七章. 重要参考文献（List of Key References）

**7.1 CFD 文献**

[1]

[2]

**7.2 DEM 文献**

EDEM商业软件的网页提供了DEM Literature Database，可以查询最新的关于离散元以及耦合模拟的理论和应用文献。链接如下：

<http://www.edemsimulation.com/resources-learning/dem-literature-database/>

以下文献是手册编写过程中引用的文献。

[1]

[2]

**7.3 Coupled CFD and DEM 文献**

[1]

[2]

# 附录：论坛问答（Q&A）