PSCALSFELES MANUAL

Michel Rasquin, Raphael Laguerre, Gérard Degrez,

Université Libre de Bruxelles, CP 165/43, 50 Av. F.D. Roosevelt, B-1050 Brussels, Belgium, mrasquin@ulb.ac.be, rlaguerr@ulb.ac.be

April 17, 2008

Abstract

SFELES is a hybrid Spectral/Finite-Element code capable of simulating fully 3D unsteady incompressible viscous flows in planar and axisymmetric geometries. SFELES runs on both sequential and distributed-memory parallel computers (using MPI).

PSCALSFELES is first of all a copy of SFELES and is thus able to do exactly the same things as mentioned just above when compiled with the usual SFELES flags and options. In this case, it can run on both sequential and distributed-memory parallel computers as well (using MPI).

When compiled with the -DPASSIVESCAL option in make.inc, PSCALSFELES becomes a tool able to simulate fully 3D unsteady incompressible viscous flows (like SFELES), but in axisymmetric geometries only (no planar version so far) and to solve some passive scalars (ike temperature, concentration,...) carried by the 3D unsteady flow. Beware! In the present implementation, it can only run on distributed-memory parallel computers (using MPI).

The next chapters will explain how to compile PSCALSFELES and how to use it with a simple test case.

Contents

Chapter 1

Introduction to PSCALSFELES

1.1 Downloading

To get started, first contact Michel Rasquin (michel.rasquin@ulb.ac.be ; + 32 650 26 90) to receive the password for access to the Université Libre de Bruxelles CVS repository on ANIC3 (mecapp45.ulb.ac.be)

Next, determine which type of shell you are using by means of the command:

>echo $SHELL

This will probably be either /bin/csh, /bin/tcsh or /bin/bash. Whenever you open one of these shells, a configuration file is executed. The name of the configuration files are, respectively, .cshrc, .tcshrc or .bashrc.

Inside the configuration file for the shell you are using, define the environment variables CVS_RSH and CVSROOT. Under csh/tcsh, add:

```
setenv CVS_RSH  ssh
setenv CVSROOT :ext:cvsusers@mecapp45.ulb.ac.be:/home/cvsusers/projects
```

Under bash, add

```
CVS_RSH=ssh
CVSROOT=:ext:cvsusers@mecapp45.ulb.ac.be:/home/cvsusers/projects
export CVS_RSH
export CVS_ROOT
```

After modifying and saving your configuration file, activate it by 'sourcing' the file:

```
>source .cshrc
```

and likewise for tcsh (source .tcshrc) and bash (source .bashrc).

You are now ready to download the software. Inside a shell, proceed as follows

```
>cvs checkout PSCALSFELES
cvsusers@mecapp45.ulb.ac.be's password:
<enter the CVS password>
>cvs checkout metis-4.0
cvsusers@mecapp45.ulb.ac.be's password:
<enter the CVS password>
>cvs checkout SuperLU_3.0
cvsusers@mecapp45.ulb.ac.be's password:
<enter the CVS password>
>cvs checkout FlexMG
cvsusers@mecapp45.ulb.ac.be's password:
<enter the CVS password>
```

Metis-4.0 is a graph partitioning program.

SuperLU_3.0 and FlexMG are general purpose libraries for the solution of large, sparse, nonsymmetric systems of linear equations on high performance machines. The library routines of SuperLU_3.0 will perform a LU decomposition with partial pivoting and triangular system solves through forward and back substitution. FlexMG rather uses an iterative procedure like GMRES or Fixed Point coupled with multigrid preconditioners. To know much about FlexMG and how to use it, please have a look at the manual in FlexMG/DOC.

Metis-4.0 and SuperLU_3.0 packages are open source projects and can be therefore be downloaded for free. It is your responsibility to read and respect the license agreements of these software packages on the web. See:

- www-users.cs.umn.edu/~karypis/metis/

- http://crd.lbl.gov/~xiaoye/SuperLU/

1.2 Compiling

The next step is to compile these libraries. Proceed as follows for SuperLU_3.0 and Metis-4.0:

```
>cd
>cd SuperLU_3.0
>make blaslib
>make superlulib
>cd
```

```
>cd metis-4.0
>make
```

Do not forget the 'make clean' command, which you can invoke for each of these softwares to remove all compiled object files (see below). If you are not an experienced programmer, then we recommend that you type 'make clean' before 'make' to make sure there are no bad object files to mess up your compilation.
For more information about the compilation of FlexMG, see the manual in FlexMG/DOC.

The compilation of PSCALSFELES follows exactly the same procedure as SFELES. Namely, in 'makefile', you will find some flags which need to be checked before compiling. As said in the abstract, PSCALSFELES s first of all a copy of SFELES.

- If you do not invoke the passive scalar flag, it will work exactly in the same way as SFELES. In this case, you can use your favorite compilation flags to run in either parallel or sequential. Beware! A bug is still creeping in SFELES when compiled in double precision! So please keep using the single precision until the code is cleaned up!.
- If you are interested in computing passive scalars and this should be the case if you manage to reach this point of the manual, you MUST compile the code with the flags -DPARALLELMPI -DSINGLEPRECISION -DPASSIVESCAL. The choice of the solver (SuperLU or FlexMG) is still up to you (-DWITHSUPERLU or -DWITHFLEXMG). For example:

  ```
  DEFINES = -DPARALLELMPI -DSINGLEPRECISION -DPASSIVESCAL -DWITHSUPERLU
  ```

  This means that you cannot run PSCALSFELES to compute a flow and a passive scalar at the same time in sequential mode! In other words, you have to use at least two active modes for the Navier-Stokes solutions and for the passive scalar solution and run the computation on two processors.

1.3 Implementation

The passive scalar equations in cylindrical coordinates are implemented in PSCALSFELES/SRC/3D/FEP1/M10_Cylindrical_NS_Galerkin. So far, there is thus no model for the cartesian coordinates. When solving the Navier-Stokes equations with a passive scalar, model M10 must then be coupled with model M04 somehow.
To illustrate the parallel implementation of PSCALSFELES, we will consider a solution in the azimuthal direction involving 16 modes. 8 active modes are computed and thus 4 processors only are required since these active modes are conjugate. Do not try to compute more than one mode per processor! Despite the fact that SFELES is able to do that, PSCALSFELES will crash!

The parallele implementation of SFELES, using 4 processors to compute 8 active modes is illustrated in Fig. 1.1.

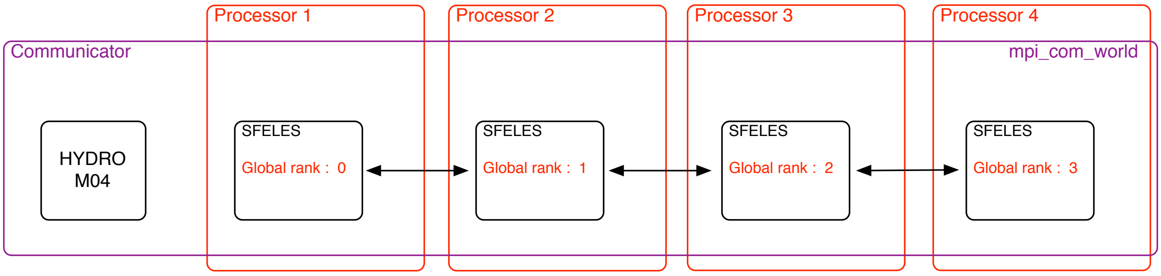


Figure 1.1: Computation of the Navier-Stokes equations - SFELES in parallel.

When computing the same flow using 8 actives modes but with two additional passive scalars for instance , we get the diagram illustrated in Fig. 1.2. Note that the passive scalar MUST have the same number of modes and actives modes!
At every time step, the solution of the Navier-Stokes equation is sent to other instances of PSCALSFELES dealing with the two passive scalars.

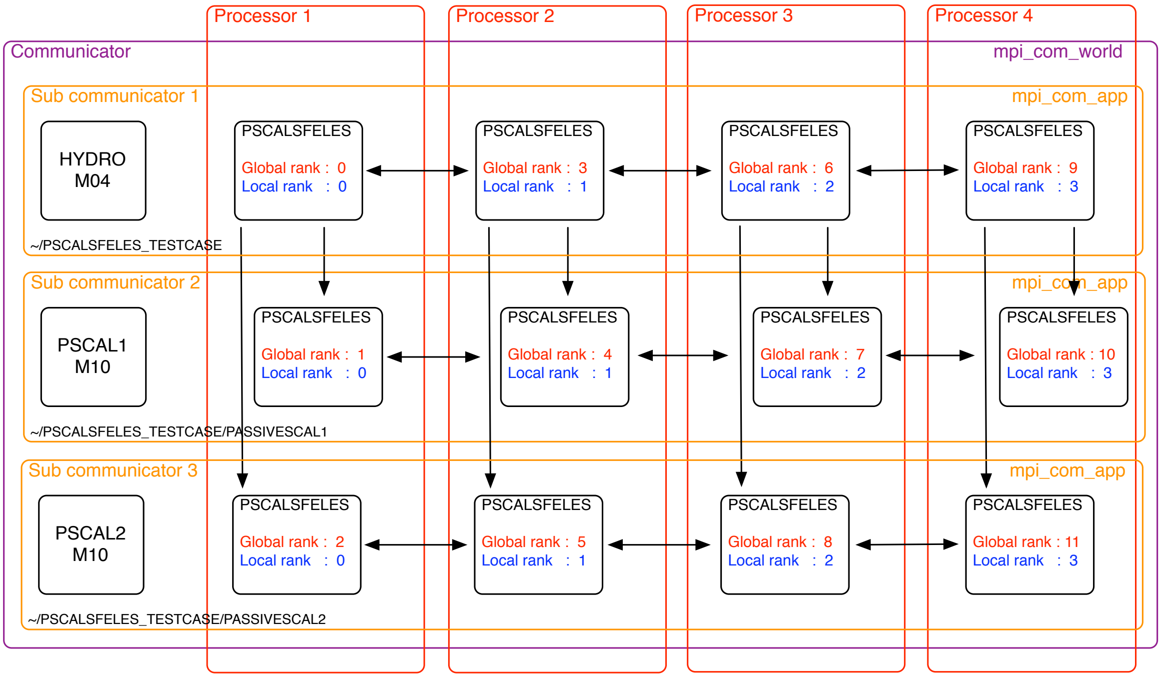


Figure 1.2: Computation of the Navier-Stokes equations and two passive scalars - PSCALSFELES in parallel.

1.4 Number of processes and processors required .

The number of processors required to run PSCALSFELES is equal to

$$nprocessors = nactivemodes/2 \tag{1.1}$$

The number of pscalsfeles instances (processes) per processor is equal to

$$nprocessesperprocessor = (1 + npassivescalars) \tag{1.2}$$

The number of pscalsfeles instances (processes) to start is thus equal to

$$nprocesses = (1 + npassivescalars) * nactivemodes/2 \tag{1.3}$$

In our example (8 active modes and two passive scalars), such a computation would be possible by invoking the following mpich2 command in the directory PSCALSFELES_TESTCASE:

```
>cd PSCALSFELES_TESTCASE/DEMONSTRATION
>mpiexec -machinefile machinefile -n 12 pscalsfeles
```

with the machinefile being for instance:

```
node5:3
node6:3
node7:3
node8:3
```

'node5:3' means 3 instances of pscalsfeles can run on node5 and so on.

Note that 12 instances of PSCALSFELES run on only 4 processors. In other words, you have to be able to run several instances of the same program on the same processor, which can be difficult on some clusters.

1.5 Differences between SFELES and PSCALSFELES

To be able to achieve such a diagram as illustrated in Fig. 1.2, some changes were brought to the original copy of SFELES named PSCALSFELES. The changes can easily be recognized since there can be found between some flags such as "!MODIF MR" or "!MODIF RL". The list of these modifications are listed in Table 1.1.

• Files added:	• Files modified:
- changedir.c	- makefile
- bc_M10.F	- main.F
- init_M10.F	- parallel.F
- linear_M10.F	- inout.F
- modules_M10.F	- modules.F
- nonlinear_M10.F	- screen.F
- output_M10.F	- vortex.F
- plotmtv_M10.F	- ftrans.F
- sgs_M10.F (empty)	- sfeles3D.F
- stats_M10.F	- init_M01.F
- stiffness_M10F	- init_M03.F
	- init_M04.F
	- stats_M04.F
	- init.F

Table 1.1: Differences between SFELES and PSCALSFELES.

Chapter 2

Test case configuration

If you have gotten this far without too much assistance, you are then definitely qualified to run your first test case. This test case will illustrate how to configure everything in order to have passive scalars being solved at the same time as the hydrodynamic Navier-Stokes equations. First, download the demonstration test case.

```
>cd
>cvs checkout PSCALSFELES_TESTCASE
>cd  PSCALSFELES_TESTCASE/DEMONSTRATION
```

2.1 Description of the test case and its directory structure

The directory structure of this test case is presented in Table 2.1 and has to be compared with Fig. 1.2. Take the time to have a look at each of the files and directories.

~/PSCALSFELES_TESTCASE/DEMONSTRATION	run_par.sh		
	machinefile		
	commandfile_m04		
	postprocessPscalsfelesHydro		
	coupeHydro_4proc.mcr		
	streamlinesHydro_4proc.mcr		
	RESULTS.tar.gz		
	GRID/	Mesh40.neu	
	SOLUTION/	control.crd	
	OUTPUT/		
	PASSIVESCAL1/	commandfile_m10_PS1	
		postprocessPscalsfelePS	
		coupePS_4proc.mcr	
		RESULTS.tar.gz	
		GRID/	Mesh40.neu
		SOLUTION/	control.crd
		OUTPUT/	
	PASSIVESCAL2/	commandfile_m10_PS2	
		postprocessPscalsfelePS	
		coupePS_4proc.mcr	
		RESULTS.tar.gz	
		GRID/	Mesh40.neu
		SOLUTION/	control.crd
		OUTPUT/	

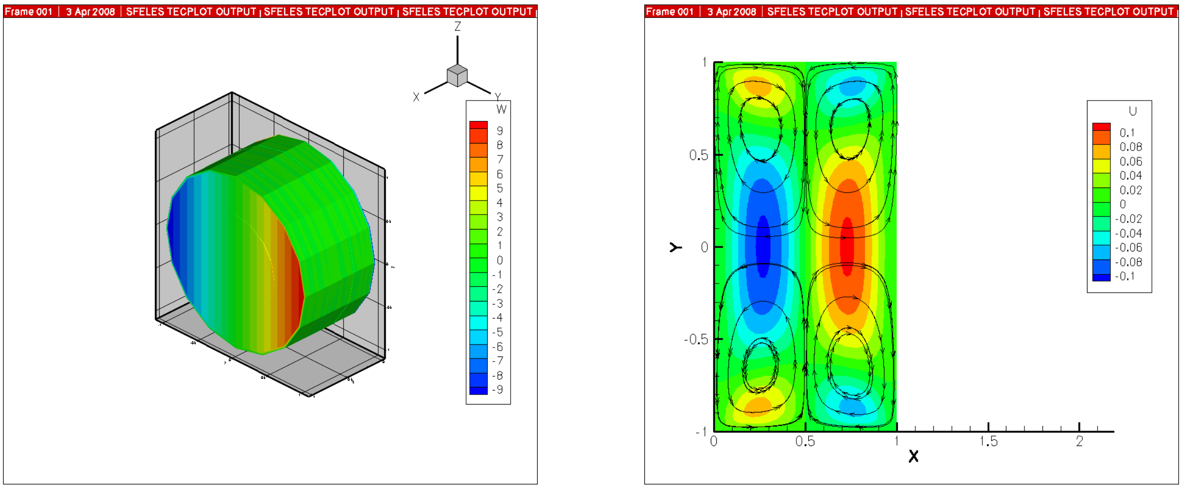
Table 2.1: Directory structure of a pscalsfeles test case.

This test case aims at solving the Navier-Stokes equations in a cylinder with its two basis rotating in counter sense. The cylindrical model M04 is used for this purpose. Two passive scalars are computed as well, using the model M10, with different boundary conditions. For both the hydrodynamic and passive scalars solutions, 16 modes - 8 active are used in the azimuthal direction.

2.1.1 Hydrodynamics

4 instances of PSCALSFELES responsible for solving the hydrodynamic equations (first sub communicator - HYDRO M04 in Fig. 1.2) run in the root directory of the test case (∼/ PSCALSFELES_TESTCASE/DEMONSTRATION). The results are stored as usual in the OUTPUT, STATISTICS and SOLUTION directories and can be postprocessed with the usual sflpost program. Have a look at the postprocessPscalsfelesHydro script. Some usefull tecplot macro are also available (*.mcr).

The results after a few time steps are presented in Fig. 2.1. An opposite sign angular velocity is thus imposed on the two basis of the cylinder while its body is kept fixed. To achieve this, 3 no slip wall conditions are imposed on the boundaries of the cylinder while the axis is treated with the special axis condition. The pressure is finally imposed in one point of the mesh.
Two big conter rotating and axisymmetric rollers can be observed. They will play a role when testing the implementation of the convective terms for the passive scalars.



(a) 3D view

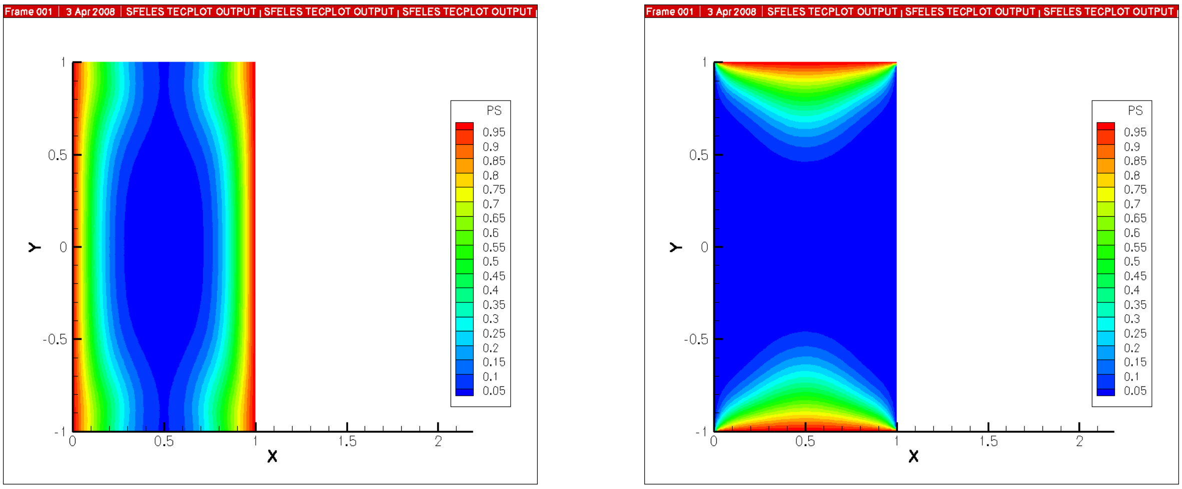
(b) 2D view (axis at Y=0 ; two cylinder basis at X=0 and X=1 ; cylinder body at $X^2+Y^2=1$)

Figure 2.1: Flow inside a cylinder with its two bases rotating in counter sense

2.1.2 Passive scalars

The other 8 instances of PSCALSFELES responsible for solving the transport equations of two passive scalars (second and third sub communicators - PSCAL1 and PSCAL2 in Fig. 1.2) respectively run in ~/PSCALSFELES_TESTCASE/DEMONSTRATION/ PASSIVESCAL1 and ~/PSCALSFELES_TESTCASE/DEMONSTRATION/PASSIVESCAL2. There solutions are again stored in OUTPUT, STATISTICS and SOLUTION, but this time in the /PASSIVESSCAL1 and /PASSIVESSCAL2 directories respectively.

The solution of the passive scalars can also be postprocessed with the usual sflpost program. Have a look at the postprocessPscalsfelePS file. Only one variable (PS) can here be post processed. Note again a usefull tecplot macro available (*.mcr). The results after the same number of time steps are presented in Fig. 2.2.



(a) First passive scalar: cylinder basis temperature fixed at 1 (no slip wall) and zero heat flux through the cylinder body (slip wall)

(b) Second passive scalar: cylinder basis temperature fixed at 0 (no slip wall) and cylinder body tempartue fixed at 1 (no slip wall)

Figure 2.2: Passive scalars results

In Fig. 2.2(a), the cylinder basis temperature (X=0 and X=1) is fixed at a constant value of 1. This correspond to a no slip wall condition. A zero heat flux is imposed on the cylinder body. At present, no entering flux can be imposed but only zero flux.
In Fig. 2.2(b), the cylinder basis temperature is fixed at a constant value of 0 while a constant temperature of 1 is imposed on the body.
In both cases, note the diffusive and convective effects on the temperature profile. The flow streamlines drawn in Fig. 2.1 must be kept in mind to explain the distortion of the temperature front presented in Fig. 2.2.

The output files used to obtain Fig.2.1 and Fig.2.2 are archived in RESULTS.tar.gz in the adequate directories. Refer to Table 2.1 to locate them. To unarchive these

*.tar.gz files, type the following command:

```
>gzip -d RESULTS.tar.gz
>tar -xvf RESULTS.tar
```

2.2 Commandfile

So far, a commandfile is required for

- the hydrodynamic equations
- for each passive scalar.

To avoid any confusion when all these commandfiles are opened in your favorite text editor, we adivse you to rename these commandfiles with an explicit name such as commandfile_m04, commandfile_m10_PS1 and so on. In your run script, just make a copy of commandfile_m** to commandfile in the adequate directories. Have a look at run_par.sh for more details.

We encourage you to have a look at these commandfiles and to compare them by making a vimdiff.

We describe here the most important parameters to be checked.

- VERBOSITY
  You will only see some messages coming from the master node solving the Navier Stokes equations. The passive scalars instances of the code will not print anything on your screen. Just play with this parameter in commandfile_m04. If verbosity is set to 1, a message will warn you whenever the solution of the Navier Stokes equations is sent to the passive scalars.
- NBFMODES and NACTIVEMODES
  The number of modes and actives mode must be the same in both commandfile_m04 and commandfile_m10_PS*.
- MUMOLEC
  The value of the viscosity (Mumolec) for the Navier-Stokes equations and for each passive scalar equation can be different. We encourage you to play a little bit with this parameters in commandfile_m10_PS* to see its influence.
- IMODEL
  It should be set to 4 to solve the Navier Stokes equations and to 10 to solve the passive scalar equations

- ALPHAC
  The default value of this parameter for the Navier Stokes equation (MO4) should be 0.5d0 in order to conserve the kinetic energy at discrete level. For the passive scalar, this value should be set to 1.d0 in order to conserve the passive scalar value in the domain. In other word, ALPHAC=1.d0 means we do not allow destruction or creation of passive scalar (concentration, pollutant) in the field.

- NPASSCAL
  This parameter set the number of passive scalar to be solved and the number of directories 'PASSIVESCAL*‘to look for. Beware of the case: the directory for the first passive scalar should be PASSIVESCAL1, the one for the second passive scalar PASSIVESCAL2 and so on. Whenever you want to compute a new passive scalar, modify NPASSCAL, add a new directory PASSIVESCAL* and modify your script and your machinefile in order to respect the rules defined in Section 1.4.

- NTSTEPS
  It is very important that the number of time steps to perform during the computation is the same for both the Navier Stokes (M04) and the passive scalars solver (M10). If it is not the case, the code will try to send some information through the network forever or wait for some information forever as well!

- DELTAT
  Keep also the same value for this parameter everywhere otherwise, your results will not be coeherent.

- GRIDFILE
  You must declare the same gridfile for all the instances of PSCALSFELES.

- INSOL, OUTSOL and IRESTART
  These parameters work for the passive scalars exactly as for the Navier Stokes equations.

- INIVAL
  For a passive scalar, only the first value will be read. You can the initialize for instance a concentration field to 1.

- Boundaries
  For the passive scalars, we can impose so far a fixed value (no slip wall) or a zero flux (slip wall) on each boundary. Like the INIVAL parameter, only the first value will be read.

Again, the best way to understand what is going on is to have a look in our PSCALSFELES_TESTCASE/DEMONSTRATION example.

2.3 Starting and running the job

The number of instances to start and the number of processors to be used has been presented in section 1.4. It is very important to follow these rules!

As said in the first chapter, the number of modes and actives modes must be the same for both the hydrodynamic and the passive scalars solutions.

Have a look at run_par.sh and machinefile. This script has to be run in the root directory of the test case (∼/PSCALSFELES_TESTCASE/DEMONSTRATION). A routine written in changedir.c and called in main.F will care about running the instances of PSCALSFELES in the adequate directories, namely the root directory of the test case for the Navier-Stokes equations (M04), root directory of the test case/PASSIVESCAL1/, for the first passive scalar equations,...)

It is important to note that restarts are allowed for the passive scalar exactly as for the Navier Stokes equations.

The probe system is also available for the passive scalars. Just add a file named control.crd in PASSIVESCAL*/SOLUTION and you will get the passive scalar value of your points in PASSIVESCAL*/OUTPUT/control.dat.

2.4 Last words

We hope this manual will be helpfull for you. If you have any question, please do not hesitate to contact us!

GOOD LUCK!