**1. DESCRIPTION**

PUReIBM-PS2D-FD1D is a three-dimensional pseudo-spectral particle-resolved direct numerical simulation solver for detailed analysis of homogeneous fixed and freely evolving fluid-particle suspensions. PUReIBM-PS2D-FD1D is a continuum Navier-Stokes solver based on Cartesian grid that utilizes Immersed Boundary method to represent particle surfaces. This code can be downloaded from

https://github.com/sciam/PUReIBM\_PS2D\_FD1D

**2. REQUIRED PACKAGES**

In PUReIBM-PS2D-FD1D, Fourier transforms are handles using the FFTW library. Therefore, it is required that this package be downloaded from "www.fftw.org" website and compiled as prior to compiling PUReIBM. It is recommended that the latest version of FFTW be used for this purpose. Note that in the serial and parallel modes, the files "fftw3.f03" and "fftw3-mpi.f90" should be used.

The parallelization of PUReIBM-PS2D-FD1D is done by two-dimensional domain decomposition technique along the x directions in the physical space.

**3. COMPILATION**

One list of file dependencies and two scripts are used to generate the makefile required for compiling PUReIBM-PS2D-FD1D. The files lists are "src\_flo\_scal" and "src\_post" that which include the name of files needed for compiling, respectively, the main code and post-processor. If any additional file is added in future to the source code, it should also be included in these file lists.

The first script that is used to generate the makefile is called "xdomake" that is the main directory. It calls another script called "domake" that is located in the folder "DOMAKE". Prior to compiling the code, it is required that the path to FFTW library directory in front of "FFTW\_LIBDIR", and also the path to MPI library directory "MPI\_LIBDIR" be changed in "domake" file. Note that we have successfully compiled and tested PUReIBM-PS2D-FD1D with Intel FORTRAN compiler (ifc). On machines where several Fortran compilers are installed, it is always safe to explicitly define the path to the compiler and linker in the "domake" file in front of "CC" and "LINK".

Once the "domake" file is modified, the makefile can be generated by running the following in the main folder:

$ ./xdomake OPTIM flo\_scal ibm.exe ifc

In the above command,

1. the first argument "OPTIM" means optimal mode. If the debug mode compilation is needed, it should be replaced with "DEBUG".
2. The second argument "flo\_scal" refers to the file list "src\_flo\_scal". If you intend to generate the makefile for post-processor, replace it with "post". If you would like to add other source files, please revise the "src\_flo\_scal" or "src\_post" also.
3. The third argument "ibm.exe" is the name of the executable file. The name can be anything you like.
4. The fourth argument "ifc" is the name of the compiler that refers to Intel FORTRAN compiler. In the "DOMAKE" folder, there is a file titles "f77\_opt\_ifc" that defines the Intel compiler and options (please open and look at the details). There are also other files that contain switches for other compilers such as gfortran, path, Portland Group.

The script also asks if the makefile for parallel simulation is needed to generate. Once the make files are made, the code is compiled by using the command line:

$ make -f MAKE/OPTIM\_ifc\_makeflo\_scal

If everything goes well, an execution file is generated in the main folder, e.g. ifc\_ibm.exe. Note that you may have the error like “cannot find -lrfftw “. This is because you did not setup the FFTW path correctly. For instance, you can open the file “OPTIM\_ifc\_makeflo\_scal” in the folder MAKE/ and setup the path as

FFTW\_LIBDIR= -L/home/your\_account\_name/FFTW\_IFC/lib/

**4. Run case**

Once the complication is done, one can use the execution file such as “ifc\_ibm.exe” to run a case as follows:

$ ./ifc\_ibm.exe test > print&

In the above command line, the argument “test” is a file name which is “test\_floparam.in”. This file contains the flow parameters that related to flow, turbulent, or heat transfer. The run\_name should be added to the beginning in "[floparam.in](http://floparam.in)" as shown above. One example is shown as follows:

-------xxxx----------------------.------------

&FLOANDGRID

NPROCY = 1 !number of processors in y direction

NPROCZ = 1………………….!number of processors in z direction

TEND = 2.000000000000000 , !not important

NERR\_STEPS = 100, !not important

WTIME\_MAXHRS = 47.5, !the maximum wall time you want to run your code. if

!you run freely evolving suspensions, this is how long

!your simulations run. For fixed assemblies, if the

!simulation converges, it stops before this time.

SAVEFORRES\_TIME = 6.d0 , !time interval for saving the restart files

TOL\_FERROR = 1.0E-06, !the tolerance for convergence criterion based on

!relative error of the average drag force

VIS = 0.012, !viscosity of the fluid. irrelevant because it is

!changed such that the mean slip velocity becomes 1

! while the Reynolds number is maintained as desired

RE = 20.0, !desired mean slip Reynolds number

RHOF = 1.000000000000 , !density of the fluid

CPF = 4179.00000000000 , !thermal heat capacity of the fluid. not relevant

!because I have not finished the heat transfer part

KF = 0.613000000000000 , !not relevant

CFL = 0.1, !the CFL number, usually set to 0.1. you can try

!higher values if you want a fast result, at

!the of expense loosing some accuracy.

RET = 0.000000000000000 , !the initial Reynolds number based on square root of granular temperature

FLO\_ANG = 0.d0, 90.d0, 90.d0 !The angles of the mean flow direction w.r.t. xyz

/

&PART\_PROPT

NBODY = 1, !irrelevant

F1 = 1.600000000000000 , !irelevant

F2 = 1.600000000000000 , !a parameter used to determine the number of

!points on a particle surface. not less than 1.6

LYBYD = 6, !length of the computational box compared to the particle diameter

DBYDX = 40, !grid resolution per particle

INPUT\_TYPE = 'random' , !!choose the case

/

&SCAL\_PROPT

SETPHIMEAN = F,

NSPMX = 1,

PHISTREAM = 1.000000000000000 , !! initial fluid temperature

PHISURF = 0.0000000000000000 , !!! surface temperature

LUMPED\_CAP = F,

PR\_OR\_SC\_NU = 0.7D0 , !! Prandtl number or Schmit number

ZERO\_FLOW\_IN\_SOLID = F,

sourcepresent = F

----------xxxx-------------------------

For example, “NBODY” represent the number of solid spheres in any simulation. More details of the parameters and the name list can be found in the source file “SRC/FLO/initialize.F90”.

Please search the source code if you have any doubt about the parameters in "[floparam.in](http://floparam.in)". It should be easy for you to understand the function of parameters.

Note that if you would like to run different cases simulation, you can choose the case using this parameter as INPUT\_TYPE = 'xx'

xx is simple (generate simple cubic array of particle configuration)

xx is random (generate random particle configuration)

xx is default (read particle configuration from input file)

There are also other different input type. Please read the source code to find out in SRC/FLO/initialize\_flo.F90 by searching “input\_type”.

5.Q & A

Any question relate to PUReIBM can be sent to [mehr@iastate.edu](mailto:mehr@iastate.edu) and [sciam@iastate.edu](mailto:sciam@iastate.edu) .