User Guide of Multiscale Complex Fluids simulation (MCF) code

Xin Bian

August 2, 2010

Contents

Abstract

1 Introduction

A code named multiscale complex fluids simulation (MCF) has been developed and is still being developed in Marco Ellero's Emmy Noether group at Prof. Adams' Chair of Aerodynamics, Technical University Munich. This article is created to facilitate usage of the code, which is written in Fortran 90 and uses a highly efficient parallel particle mesh (PPM)[1] library. The numerical algorithme of MCF is based on mesh free Lagrangian particle method, particularly smoothed particle hydrodynamics (SPH) and its extended version smoothed dissipative particle dynamics (SDPD).

2 Compiling

2.1 Parallel particle mesh library

`http://chaton.inf.ethz.ch/ppm/index.php/Main_Page` [1].

2.2 Multiscale complex fluids simulation

2.3 Punto

A easy use visulization tool.

2.4 Paraview with particles

3 Input configuration files

There are three input configuration files located in *$/mcf/mcf_config/* directory, i.e., **mcf_ctrl_config_file**, **mcf_physics_config_file** and **mcf_io_config_file**. Their default names are *ctrl.mcf*, *physics_config.mcf* and *io_config.mcf*. For people who are familiar with SPH or even general Lagrangian particle method, such as molecular dynamics (MD), there should be no difficulty in using MCF code following the comments in these three input files. To have a complete explaination, we describe all the parameters here. Maybe some of the text here are redundant, since it is basically an extended version of the comments of the three input files. Parameters' names are always in **bold** and their values are always in *italic*.

3.1 mcf_ctrl_config_file (*ctrl.mcf*)

Before running the simulation, this file should be always the first file checked by the user. It includes definitions of the fundamental components of MCF. The name *ctrl.mcf* is hard coded in MCF. If one wants to use another control file *FILE*, he can start MCF with a parameter in command line, i.e., $ ./mcf *FILE*. Note *FILE* must include all the parameter explained below.

- **mcf_io_config_file**
  File name of IO information input file.
  Default is *io_config.mcf*.

- **mcf_physics_config_file**
  File name of physics information input file.
  Default is *physics_config.mcf.*

- **debug_flag**
  If the simulation runs in debug mode (MCF compiled with debug flag), this parameter determines the level of debug, the higher, the more extensive. Default is *1*.

- **relax_run**
  If particles' positions needed to be relaxed or randomized from the initial lattice, set this value to *TRUE*, then MCF will use SDPD to pre-run. In this case, **relax_type** etc. in **mcf_physics_config_file** file should be set accordingly.

- **read_external**
  If MCF needs to read particles' configurations externally, this value is set to be *TRUE*; It includes particles' configuration *(x, v, rho, m, p_id, s_id)* and conformation tensor(for Non-Newtonian oldroyd-B fluid). IO parameters in **mcf_io_config_file** file have to be set accordingly.

  For *FALSE*, particles are completely generated by MCF.

- **rhs_density_type**
  Formulation of right hand side (RHS) for density.

  - *1* Summation formulation of physical density[2].
  - *2* Summation formulation of number density[3].
  - *3* Integration formulation of physical density[2][4] in continuity equation (not available).

- **rhs_force_type**
  Formulation of RHS for force.

  - *1* Morris J.P. et al. [4]
  - *2* Español P. and Revenga M. [3]
  - *3* Hu XY and Adams N. A. [5].

- **dynamic_density_ref**
  To subtract backgroud pressure, one needs to use reference density. If this variable is set to be *TRUE*, the reference density will be calculated dynamically, which means ρ_{ref} is always the minimum density at every step. This should not be used unnessarily, since it needs communication between processors to determine minimum density.

  For *FALSE*, ρ_{ref} will be given in **mcf_physics_config_file** file.

- **stateEuqation_type**
  Type of state equation.

  - *1* $p = c_s^2(\rho - \rho_{ref})$ [4]
  - *2* $p = p_0\left(\left(\frac{\rho}{\rho_{ref}}\right)^{\gamma} - 1\right)$ [6]

- **kernel_type**
  Type of kernel being used.
  - *1* Quintic Spline [4].
  - *2* Lucy kernel [7].
  - *3* B-Spline (not available).
- **integrate_type**
  Time integration scheme.
  - *1* Explicit Euler.
  - *2* Modified velocity Verlet [8].
- **symmetry**
  Set this to *TRUE* if use symmetry inter-processor communiction and inter-particle interaction; *FALSE* otherwise.
- **flow_v_fixed**
  TRUE: In-flow velocity is kept as a constant value by adjusting body force dynamically. Set **flow_direction** etc. parameters in **mcf_io_config_file** file.
- **Brownian**
  TRUE: Brownian motion will be considered under certain temperature, **kt** should be set in **mcf_io_config_file** file. Currently Brownian motion only works for **rhs_density_type** = *2, 3*
- **Newtonian**
  Logical value indicating if we are simulating Newtonian or Non-Newtonian fluid (e.g. viscoelastic oldroyd-B model). Currently it only works for **rhs_density_type** = *2, 3*.
- **p_energy**
  Logical value indicating if potential energy needs to be calculated.
- **write_restart**
  Write particles' information during simulation for restarting (logical variable). Set **write restart** paramters accordingly in **mcf_io_config_file** file.

3.2 mcf_physics_config_file (*physics_config.mcf*)

This files contains the physical parameters for the simulation.

⟨complex fluids ⟩

- **num_species**
  If set to *1*, colloids will be ignored, there is only fluid; If set to *2*, colloids must be given later. Remark : wall boundaries are **not** considered as another different species.

⟨space ⟩

- **num_dim**
  Space dimensions. Either *2* or *3*.

- **min_phys**
  Bottom boundary coordinates *min_x, min_y (, min_z).*

- **max_phys**
  Top boundary coordinates *max_x, max_y (, max_z).*

- **lattice**
  Type of lattice for generating particles configuration.

  - 2D : *1* square; *2* staggerd (not available); *3* hexagonal.
  - 3D : *1* simple cubic lattice; *2* body centered lattice (not available); *3* face centered lattice (not available).

- **num_part**
  Initial estimated number of particles in x, y(, z) directions, supposing particles are on lattice.

- **cut_off**
  cut_off of compact support domain r_c.

⟨time scale ⟩

- **dt**
  Time step. If negative value is given, dt will be calculated according to CFL, viscous diffusion, and magnitude of particle accerleartion [4] in the simulation.

- **step_start**
  Number of time step to start.

- **step_end**
  Number of time step to end. However, the bigger one of **step_end** × **dt** and **time_end** will be chosen (**dt** is time step used in the simulation).

- **time_start**
  Time to start.

- **time_end**
  Time to end.

⟨physical properties ⟩

- **rho**
  Initial density of solvent.

- **eta**
  Absolute/dynamic (shear) viscosity.

- **ksai**
  Bulk viscosity used in Español and Revenga [3]'s formulation for compressible flow (not available).

- **kt**
  Boltzmann constant × Temperature, used for Brownian motion in SDPD.

- **c**
  Artificial sound speed.

- **rho_ref**
  Reference density ρ_{ref}; If dynamic density referece is required ρ_{min} will be calculated in the simulation and taken as reference density.

- **gamma**
  Exponenet in the equation of state.

⟨relax run/pre-run parameters ⟩

- **relax_type**
  Relaxation criterion:
  - *1* according to given step/time.
  - *2* according to desired disorder level.

- **step_relax**
  Steps for relax run.

- **time_relax**
  Time for relax run.

- **disorder_level**
  Desired disorder level [*0, 1*].
  0: completely disorder; *1*: on lattice;

- **kt_relax**
  Boltzmann constant × Temperature for SDPD pre-run.

- **c_relax**
  Sound speed for relax run.

⟨external force ⟩

- **body_force_type**
  Type of body force
  - *0* no body force.
  - *1* constant in one direction.
  - *2* constant in two direction.
  - *3* $F_0 \times sin(k \times y)$.

- **body_force**
  Body force F_0 applied to all solvent particles.

- **body_force_d**
  Increment of body force, when fixed flow velocity strategy is used.

⟨flow velocity ⟩

- **flow_direcion**
  Direction of desird flow.

- **flow_width**
  The box width to calculate the flow velocity. [0,**flow_width**]

- **flow_v**
  Imposed average flow velocity for [0,**flow_width**], when *TRUE* is set to **flow_velocity_fixed** in **mcf_ctrl_config_file** file.

- **flow_adjust_freq**
  The frequency to adjust bodyforce to get expected flow velocity.

⟨non-Newtonian viscoelastic Oldroyd-B parameters ⟩

- **tau**
  relaxation time of polymer molecules.

- **n_p**
  number of dumbells per unit volume.

- **kt_p**
  Boltzmann constant × Temperature for dumbell.

- **eigen_dynamics**
  Logical value indicating if we are using eigen-dynamics way.

- **eval**
  Initial eigenvalues for eigen-dynamics.

- **evec**
  Initial eigenvectors for eigen-dynamics. array notation of 2D in order :

⟨Boundary condition of physical domain ⟩

- **bcdef**
  Boudary condition definition for different sites

  - 2D: $x_{min}, x_{max}, y_{min}, y_{max}$.
  - 3D: $x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}$.

  Theire values are integer numbers

  - *1* periodic boundary condition.
  - *3* symmetry boundary condition. **symmetry** must be set to *FALSE* in **mcf_ctrl_config_file** file.
  - *7* wall boundary, outside computational domain, handeled by PPM using symmetry/mirror boundaries. **symmetry**=*FALSE* must be set in **mcf_ctrl_config_file** file.
  - *9* solid wall boundary, outside of initial computational domain created by MCF extending computational domains.
  - *10* Leeds-Edwards boundary.

- **shear_type**
  Type of shear:
  - *1* normal.
  - *2* oscillating with cosine function.
- **shear_v**
  Initial magnitude of shear velocity for different sides.
  - 2D: shear velocity of x side(left, right), in y direction; shear velocity of y side(down, up), in x direction;
  - 3D: shear velocity of x side(left, right), in y-z direction; shear velocity of y side(down, up), in x-z direction; shear velocity of z side(down, up), in x-y direction;
- **shear_freq**
  Frequecy of oscillating shear.
- **wall_rho**
  Density type of wall:
  - *0* constant.
  - *1* calculated and renormalized.
- **wall_noslip**
  Order for different walls
  - 2D: $W_{x1}, W_{x2}, W_{y1}, W_{y2}$;
  - 3D: $W_{x1}, W_{x2}, Wy1, W_{y2}, W_{z1}, W_{z2}$;

  No slip type is an integer number:
  - *1* frozen
  - *2* Morris et al. 1997 [4]

⟨colloid's parameters ⟩

Note : The order of parameters are fixed here, which should be followed.

- **num_colloid**
  Number of colloids.
- **colloid_rho**
  Density type :
  - *0* constant.
  - *1* calculated.
- **coll_x**
  Initial geometrical center of the colloid.
- **coll_translate**
  Indicate if the colldoid can translate freely.

- **coll_v**
  Initial translating velocity of center of the colloid.

- **coll_rotate** Indicate if the colldoid can rotate freely.

- **coll_omega**
  Initial rotating velocity of the colloid, always 3D vector.

- **coll_shape**
  Shape of the colloidal particle:

  - *1* disk (2D)/sphere (3D).
  - *2* ellipse (2D)/ellpsoid (3D).
  - *3* arbitrary (2D).

- **coll_ra**
  Initial (major) radius of the colloid.
  (For dis/sphere, used as radius.)

- **coll_rb**
  Initial (minor) radius of the colloid.
  (For ellipse (2D), used as minor axis; for arbitray (2D), used as fluctuating magnitude.)

- **coll_phi**
  Initial angle between major axis and $x+$ coordinate.

- **coll_freq**
  Useful for arbitray shape, determining the roughness/frequency. Must be integer.

- **coll_m**
  Mass of the colloid, not needed to be given and will be calculated according to its volume and density. Should be given for a restarted run.

- **coll_mmi**
  Mass momentum inertia of the colloid, not needed to be given and will be calculated according to its distribution of mass. Should be given for a restarted run.

- **coll_pace**
  Type of placement for colloidal boundary particles:

  - *1* on lattice.
  - *2* parallel to the surface, with fixed distance on each layer.
  - *3* parallel to the surface, with fixed number of particles on each layer.

- **coll_noslip**
  No slip type for the colloidal surface:

  - *1* frozen.
  - *2* Morris et al. 1997.
  - *3* Zhu et al. 1999.

- **coll_body_force**
  Body force on colloid.

3.3 mcf_io_config_file (*io_config.mcf*)

⟨particles configuration, if reading from external files ⟩

- **read_particles_file**
  Name of particles file for reading.
- **read_particles_fmt**
  Format of the file.
  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)
- **read_conformation_file** (non-Newtonian Viscoelastic fluids)
  Name of conformation tensor file for reading.
- **read_conformation_fmt** (non-Newtonian Viscoelastic fluids)
  Format of the file.
  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)

⟨output relax run/pre-run particles ⟩

- **output_relax_particles_file**
  Prefix name of ouput files. *timestep.out* will be appended.
- **output_relax_particles_fmt**
  Format of output file.
  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)
- **output_relax_particles_freq**
  Frequency for writing output files (in computer time steps).

⟨output particles results ⟩

- **output_particles_file**
  Prefix name of ouput files. *timestep.out* will be appended.
- **output_particles_fmt**
  Format of output file
  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)
- **output_particles_freq**
  Frequency for writing output files (in computer time steps).
- **output_conformation_file**
  Prefix name of conformation files. *timestep.out* will be appended.
- **output_conformation_fmt**
  Format of conformaiont file.

  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)

- **output_conformation_freq**
  Frequency for writing conformation files (in computer time steps).

⟨output relax run/pre-run statistics ⟩

- **statistic_relax_file**
  Name of statistics output file.

- **statistic_relax_fmt**
  Format of statistics output file.

  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)

- **statistic_relax_freq**
  Frequency for writing statistics output (in computer time steps).

⟨output statistics ⟩

- **statistic_file**
  Name of statistics output file.

- **statistic_fmt**
  Format of statistics output file.

  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)

- **statistic_freq**
  Frequency for writing statistics output (in computer time steps).

⟨output boundary information, if there is wall⟩

- **boundary_file**
  Prefix of name of boundary output file.

- **boundary_fmt**
  Format of boundary output file.

  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)

- **boundary_freq**
  Frequency for writing boundary output (in computer time steps).

⟨output colloid(s) information, if there is colloid⟩

- **colloid_file**
  Prefix of name of colloid output file.

- **colloid_fmt**
  Format of colloid output file. One of

  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)

- **colloid_freq**
  Frequency for writing colloid output (in computer time steps).

⟨write information for restarting ⟩

- **restart_particles_relax_file**
  Prefix of the name of restart file for particles *time step.dat* will be appended.
- **restart_particles_relax_fmt**
  format of restart particles file:
  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)
- **restart_relax_freq**
  Frequency for writing restart files in computer time steps. (physics and particles have same frequencey)
- **restart_physics_file**
  Prefix of the name of restart file for physics *time step.dat* will be appended.
- **restart_physics_fmt**
  Format of read physics file:
  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)
- **restart_particles_file**
  Prefix of the name of restart file for particles *time step.dat* will be appended.
- **restart_particles_fmt**
  Format of restart particles file:
  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)
- **restart_conformation_file**
  Prefix of the name of restart file for conformation tensory, *time step.dat* will be appended.
- **restart_conformation_fmt**
  Format of restart conformation file:
  - FORMATTED (ASCII text)
  - UNFORMATTED (binary)
- **restart_freq**
  Frequency for writing restart files in computer time steps. (physics and particles have same frequencey)

4 Running simulation

4.1 Local workstation

4.1.1 Serial run

One can start the simulation with default control configuration file by simply typing in command line

:> **./mcf**

then file **mcf_ctrl_config_file**=*ctrl.mcf*, which is located in the same directory, will be used as control configuration file. The name (*ctrl.mcf*) is hard-coded in MCF. In this case, **mcf_physics_config_file** and **mcf_io_config_file** must be defined inside *ctrl.mcf* also, to specify physics and IO configuration files.

If one changed the value of **mcf_ctrl_config_file**, i.e., change the name of *ctrl.mcf* to *FFFF*, he can start the simulation by

:> **./mcf** *FFFF*

Now *FFFF* is the new control configuration file, which must also define **mcf_physics_config_file** and **mcf_io_config_file**.

Since the simulation will print out the component details/parameters onto standard output (screen) while running, one can use

:> **./mcf** | **tee** *mcf_std.out*

to save the print-out into file *mcf_std.out* also, which can be used later for reference.

4.1.2 Parallel run

Make sure **mpd** is running in background as a dameon for MPI applications. Then start **mcf** in the same way as before, except that give extra parallel information of using multiple processes

:> **mpiexec -n** *NUM* **./mcf** | **tee** *mcf_std.out*

NUM is the number of processes being used.

4.2 Altix 4700 Supercomputer on LRZ

Since PBS system is used on supercomputer, one must edit a script decribing the job before actually run any simulation. The typical script is following:

```
#!/bin/bash
#PBS -o /ptmp1/pr95zi/lu64yuz2/mcf_run/standard.out
#PBS -j oe
#PBS -S /bin/bash
#PBS -l select=256:ncpus=1:mem=500mb
#PBS -l walltime=48:00:00
```

```
#PBS -N lub3D
#PBS -M xin.bian@aer.mw.tum.de
#PBS -m abe
. /etc/profile.d/modules.sh
cd /ptmp1/pr95zi/lu64yuz2/mcf_run/
mpiexec -n 256 ./mcf > mcf_std.out
```

For more details, please check `http://www.lrz.de/services/compute/hlrb/`

5 Output data files

5.1 output_particles_relax_file (*mcf_init_particlesNNN.out*)

NNN represent the number of steps in the simulation for this output file. This file contains the fundamental quantities of each particle at the end of step *NNN*. $[x, y(, z), v_x, v_y(, v_z), rho, m, p_{id}, s_{id}]$. Note that *rho* can be number density or mass density, depending on the **rhs_density_type**. Both p_{id} and s_{id} are integer numbers. p_{id} is particle ID and unique for each particle. s_{id} is species ID.

For solvent particle s_{id}*=0*. For colloidal boundary particle s_{id}*=1...MM*. *MM* is the total number of colloids in the simulation. For wall boundary particle s_{id}*=-1, -2, -3, -4,(-5, -6)*, denpending on which wall it belongs to.

5.2 output_particles_file (*mcf_particlesNNN.out*)

NNN represent the number of steps in the simulation for this output file. This file contains the fundamental quantities of each particle at the end of step *NNN*. $[x, y(, z), v_x, v_y(, v_z), rho, m, p_{id}, s_{id}]$. Note that *rho* can be number density or mass density, depending on the **rhs_density_type**. Both p_{id} and s_{id} are integer numbers. p_{id} is particle ID and unique for each particle. s_{id} is species ID.

For solvent particle s_{id}*=0*. For colloidal boundary particle s_{id}*=1...MM*. *MM* is the total number of colloids in the simulation. For wall boundary particle s_{id}*=-1, -2, -3, -4,(-5, -6)*, denpending on which wall it belongs to.

5.3 output_conformation_file (*mcf_conformationNNN.out*)

NNN represent the number of steps in the simulation for this output file. This file contains the fundamental quantities of each particle at the end of step *NNN*. $[x, y(, z), v_x, v_y(, v_z), rho, m, p_{id}, s_{id}]$. Note that *rho* can be number density or mass density, depending on the **rhs_density_type**. Both p_{id} and s_{id} are integer numbers. p_{id} is particle ID and unique for each particle. s_{id} is species ID.

For solvent particle s_{id}*=0*. For colloidal boundary particle s_{id}*=1...MM*. *MM* is the total number of colloids in the simulation. For wall boundary particle s_{id}*=-1, -2, -3, -4,(-5, -6)*, denpending on which wall it belongs to.

5.4 statistic_relax_file (*mcf_init_statistic.dat*)

For relax run/ pre-run statistics.
[step, time, kinetic energy].

5.5 statistic_file (*mcf_statistic.dat*)

Statistics file of simulation.
[step, time, kinetic energy].

5.6 boundary_file (*mcf_boundary.dat*)

Boundary (wall) file of simulation.
[step, time, kinetic energy].

5.7 colloid_file (*mcf_colloidM.dat*)

M is number index of a colloidal particle. MM is the total number of colloidal particles.
$step, time, [x, y(, z)], [v_x, v_y(, v_z)], F_x, F_y(, F_z), [\phi], [(\omega_x, \omega_y(, \omega_z)], [T_x, T_y(, T_z)]$.

5.8 restart_physics_file (*mcf_restart_physicsQQQ.mcf*)

5.9 restart_particles_file (*mcf_restart_particlesQQQ.dat*)

5.10 restart_conformation_file (*mcf_restart_conformationQQQ.dat*)

References

[1] I. F. Sbalzarini, J. H. Walther, M. Bergdorf, S. E. Hieber, E. M. Kotsalis, and P. Koumoutsakos. Ppm - a highly efficient parallel particle-mesh library for the simulation of continuum systems. *J. Comput. Phys.*, 215(2):566–588, 2006.

[2] J. J. Monaghan. Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astrophys.*, 30(1):543–574, 1992.

[3] P. Español and M. Revenga. Smoothed dissipative particle dynamics. *Phys. Rev. E*, 67(2):026705, 2003.

[4] J. P. Morris, P. J. Fox, and Y. Zhu. Modeling low reynolds number incompressible flows using sph. *J. Comput. Phys.*, 136(1):214–226, 1997.

[5] X. Y. Hu and N. A. Adams. A multi-phase sph method for macroscopic and mesoscopic flows. *J. Comput. Phys.*, 213(2):844–861, 2006.

[6] J. J. Monaghan. Simulating free surface flows with sph. *J. Comp. Phys.*, 110:399–406, 1994.

[7] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astron.J.*, 82(12), 1977.

[8] R. D. Groot and P. B. Warren. Dissipative particle dynamics: Briding the gap between atomistic and mesoscopic simulation. *J. Chem. Phys.*, 107(11):4423–4435, 1997.