TBFsolver user guide

Paolo Cifani

Example of turbulent bubble-laden channel flow

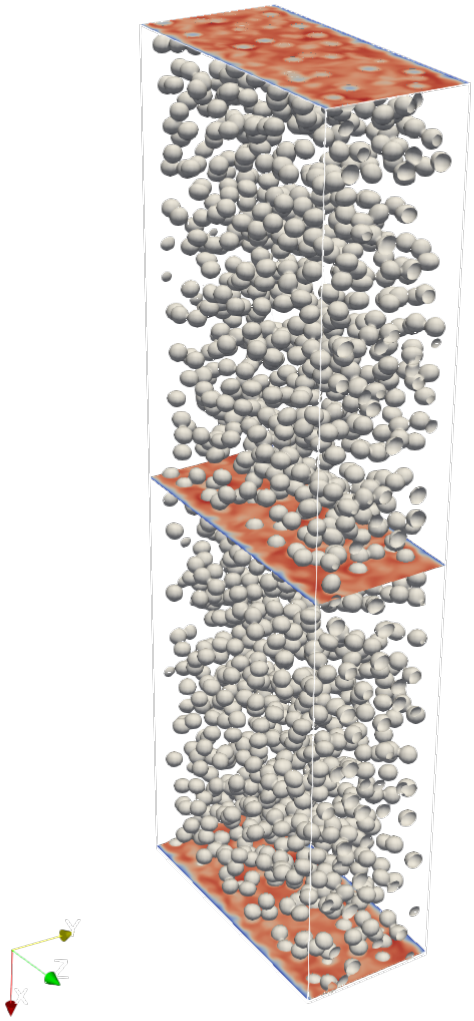

Figure 1: Contour levels of the stream-wise velocity component for different slices a of bubbly channel flow at shear Reynolds number 127. The deformable bubbles are represented by grey objects, which are the computed isosurfaces $f = 0.5$ (with f the gas volume fraction).

Structure of a test case

- 0 **folder**: contains the internal field values (in binary form) of the primary variables of the problem. These fields must be read from the solver at the beginning of the simulation. The boundary conditions and the reading options have to be set in the corresponding files in folder 'fstart_bc' (see explanation below). Note, this folder may also be empty in which case the fields will be initialised according to the setting of the corresponding files in folder 'fstart_bc'.

  The boundary conditions are specified by the following integer values:

  - type 1: fixed value
  - type 2: fixed normal gradient
  - type 3: periodic
  - type 6: hydrophobic-like contact angle (only used for the volume fraction field)

  Boundary type 4 and 5 are used internally by the solver.

- **specs folder**: contains the parameters of the simulation. Possible user entries are: integer, real (double precision) and logical. Each entry is specified by an identification name followed by one blank space and its value. Note: do not use a 'tab' or more than one blank space to separate the parameter name by its value.

  List of input files:

  - decompose:
    - `px` (integer): number of cores along the x-direction
    - `py` (integer): number of cores along the y-direction
    - `pz` (integer): number of cores along the z-direction
    - `wrap_x` (logical): TRUE if the x-direction is periodic; FALSE otherwise.
    - `wrap_y` (logical): TRUE if the y-direction is periodic; FALSE otherwise.
    - `wrap_z` (logical): TRUE if the z-direction is periodic; FALSE otherwise.
  - initBubbles:
    - `method` (integer): three different methods to initialise bubbles are available:
      - 1: single bubble
        - `R` (real): bubble radius
        - `x0` (real): initial bubble position along x-direction
        - `y0` (real): initial bubble position along y-direction
        - `z0` (real): initial bubble position along z-direction
        - `nref` (integer): number of grid refinement to compute the initial bubble field. The higher the number the more accurate the initial volume fraction field will be.
      - 2: array of bubbles
        - `R` (real): bubbles radius
        - `nix` (integer): number of bubbles along the x-direction
        - `nby` (integer): number of bubbles along the y-direction

    - `nbz` (integer): number of bubbles along the z-direction
    - `nref` (integer): number of grid refinement to compute the initial bubble field. The higher the number the more accurate the initial volume fraction field will be.
    - `random_distr` (logical): if TRUE an initial random distribution of the bubbles is set. If FALSE the bubbles are uniformly distributed along the cartesian directions.
  - 3: two bubbles. Same parameters as for one bubble. The parameters of the two bubbles are identified by `_b1` and `_b2`.
- mesh
  - `Lx` (real): length of the domain along the x-direction
  - `Ly` (real): length of the domain along the y-direction
  - `Lz` (real): length of the domain along the z-direction
  - `isXunif` (logical): TRUE sets the grid distribution along the x-direction to be uniform; FALSE sets the grid distribution to follow an hyperbolic tangent profile.
  - `isYunif` (logical): TRUE sets the grid distribution along the y-direction to be uniform; FALSE sets the grid distribution to follow an hyperbolic tangent profile.
  - `isZunif` (logical): TRUE sets the grid distribution along the z-direction to be uniform; FALSE sets the grid distribution to follow an hyperbolic tangent profile.
  - `nx` (integer): number of grid points along the x-direction
  - `ny` (integer): number of grid points along the y-direction
  - `nz` (integer): number of grid points along the z-direction
- parameters
  - `rhol` (real): liquid mass density.
  - `rhog` (real): gas mass density.
  - `mul` (real): liquid dynamic viscosity.
  - `mug` (real): gas dynamic viscosity.
  - `sigma` (real): surface tension.
  - `gCH` (real): gravity force acting along the x-direction in the vertical channel flow configuration.
  - `g` (real): gravity force.
- pcg_solver
  - `nLevels` (integer): number of multigrid levels.
  - `nPreeSweep` (integer): number of pre-smoothing sweeps.
  - `nPostSweep` (integer): number of post-smoothing sweeps.
  - `tolMG` (real): multigrid solver tolerance.
  - `maxIterMG` (integer): maximum number of iterations of the multigrid solver.
  - `fullInfoMG` (logical): TRUE: output of each multigrid solver iteration. FALSE: output only converged multigrid solver iteration.

  - `tolPCG` (real): conjugate gradient solver tolerance.
  - `maxIterPCG` (integer): maximum number of iterations of the conjugate gradient solver.
  - `fullInfoPCG` (logical): TRUE: output of each conjugate gradient solver iteration. FALSE: output only converged conjugate gradient solver iteration.
- schemes
  - `time_scheme` (integer): time integration scheme of the momentum equation:
    - 0: Adams-Bashforth 2.
    - 1: Runge-Kutta 3.
  - `convection_scheme` (integer): momentum convective term discretisation schemes:
    - 0: central differencing scheme.
    - 1: upwind scheme.
    - 2: QUICK scheme.
    - See others described in the scheme file.
- flowControl
  - `flowCtrl` (integer):
    - 1 = fixed pressure gradient. The target Re_τ has to be specified with the entry Ret.
    - 2 = fixed mass flow rate.
    - 3 = balanced gravity force when periodic boundary conditions are imposed along the direction of gravity.
- initVelocity:
  - `perturbed_parabolic` (logical): TRUE: initialises the velocity field to a perturbed parabolic profile (only used for channel flows). It set to TRUE, then parameter `iCA` should be set to the target velocity value along the stream-wise center-line of the channel.
- flowSolver
  - `flow_solver` (integer):
    - 1 = single-phase flow solver is used.
    - 2 = two-phase flow solver is used.
- timeControl
  - `Tf` (real): final simulation time.
  - `dt` (real): time-step size.
  - `writeInterval` (integer): an output folder is written every `writeInterval` number of time-steps.
  - `dtout` (real): an output folder is written every `dtout` time units. If you use this parameter, set `writeInterval` to -1.
  - `input_folder` (integer): number of folder to start the simulation from.
  - `setTimeStep` (logical): TRUE: set the time-step to the minimum value required by:

    - stability of the viscous term
    - the capillarity waves
    - the maximum CFL number defined by parameter `cflMax`.
  - `adaptiveTimeStep` (logical): TRUE: the time-step size will be adjusted according to `cflMax`. FALSE: a constant time-step size is imposed.
  - `cflMax` (real): maximum CFL number of the simulation. It works only when `adaptiveTimeStep` is set to TRUE.
  - `process_averages` (logical): TRUE: statistical quantities post-proc activated.
  - `Ts` (real): staring time of statistical averaging.
  - `Tr` (real): ramp-up interval (in time units) of the material properties.
  - `isRampUp` (logical): TRUE: the material properties will be ramped-up from time 0 to `Tr`. FALSE: the material property are assigned to a fixed value as prescribed in the file 'parameters'.
  - `vofBlocksRedInterval` (real): time interval (in time units) of the block redistribution routine.
  - `restart_boxes` (logical): TRUE: read boxes from disk; FALSE: initialise boxes. (note: if input_folder>0 the boxes are automatically read in).

- **fstart_bc folder**: contains the input fields of the primary variables of the problem: volume fraction c, pressure psi, velocity u. These files specify the initial and boundary conditions:

  - `grid_type` (character): cl = collocated grid arrangment, sx = staggered grid arrangment along x, sy = staggered grid arrangment along y, sz = staggered grid arrangment along z.
  - `init_opt` (integer):
    - 1: initialise field to a uniform value specified by parameter `iv`.
    - 2: read internal field from file.
  - `bi` (integer): with $i = 1...6$ indicated the six boundaries of the domain ordered as left, right, bottom, top, back, front. Each number specify a different boundary conditions. Valid entries for the boundary conditions are:
    - type 1: fixed value
    - type 2: fixed normal gradient
    - type 3: periodic
  - `bv` (real): specify the value of the boundary condition. When a prescribed value is not required (e.g. periodic boundaries) set `bv` to its default value $0.d0$, which will be eventually ignored by the solver.

  For velocity u the above entries `bi` and `bv` are specified with x, y and z indicating the x, y and z component of the velocity field.