BndBlock.cpp

Constructor: `BndBlock bem(number of boundaries);`

Initialize a Boundary: `Elements[id].Init(Number Of Panels, Max. Number Of Elements, Boundary Type);`

`boundary type` = 0: solid wall, 1: moving wall, 2: close liquid interface, 3: wall bound liquid interface

Add a panel: `PanelInit(id,number of points, geometry type, boundary condition type, b.c. values, geo values);`

`geometry type` = 0: straight line, between geo values (x1,y1 -> x2,y2), 1: curved segment between (x1,y1 -> x2,y2 with radius r, 2: circle, x0,y0, r and perturbation amplitude a and wave number k, 3: ellipse with x0,y0 and half axis in x and y, 4: rectangle with rounded ends with length in y-direction and width in x-direction

`bound. cond. type` = 0: no-slip wall, 1: normal/tangential velocity, 2: normal stress, 3: Poiseuille mean velocity, 4: x,y velocity, 5: x,y velocity at infinity, 6: point source, 9: pressure

Enable writing: `EnableWrite(`directory`);`

Load drop shape: `DropRead(id, timestep);`

`id` is the id number of the Element but has to fit also the id of the file, timestep is also the timestep of the file.

Move a panel: `Translate(id, x-shift, y-shift, radian turn);`

Autom. Remesh: `RemeshOn(desired element length, 0);`

Automatic remeshing works only in RK loops, call *bem*.**SeedDrop**( ); in your loop.

Correct area: `bem.Elements[boundary id].initialArea = bem.getArea(boundary id);`

Area and position: `getArea(Id);` or `getCoM(Id,X,Y);`

The functions return the area and getCoM returns the position on the center of mass to the variables.

Log Data: `LogAreaPos(File Id, Object Id);` or `LogTimeStep(File Id);`

Create a file `LogId.txt` and save area and center of mass or the Time Step and nondimensional time in the simulation.

Data Structure:

`Elements[Id].Panels;`	Number of panels for this boundary id
`Elements[Id].PanelSize[Panel];`	Number of points for this panel
`Elements[Id].XId[Panel][Point];`	Exists also as YId
`Elements[Id].BCtype[Panel];`	See above bond. cond. type number code
`Elements[Id].BCvalueX[Panel];`	Exists also as BCvalueY

Bold designates fixed names in Ulambator, *Italic* user defined names and Roman are variables, with minuscules double, staring with a majuscule integer and in 'parenthesis' char arrays.

MatrixBlock.cpp

Constructor: `MatBlC1 mat(&bem, aspect ratio, viscosity ratio, capillary number);`

&bem passes a pointer to the bndblock information, capillary number is only a suggested value that appear in **LogCfg** and has no further influence.

Enable writing: `EnableWrite(`directory`);`

Enables writing for BndBlock as well.

Create a subfolder: `SubFolder(`folder name`);`

If the folder exists the program aborts in order not to overwrite data.

Set a time step for RK: `setTimeStep(delta T, number of iterations when calling SolveRK#);`

`Perform several iterations:` `SolveRK1( );` or `SolveRK2( );`

`Build and Solve once:` `Build( delta T);` and `Solve( );`

The timestep delta T is used for stabilization and should correspond to the advanced time.

Solid problem inversion: `PreCondense( );`

For a high ratio of wall/liquid unknowns the wall problem can be precondensed. Field data export will no longer work!

Field Output: `VTKexport/VisMatlab(Time Step, Resolution, left margin, right, bottom top);`

Resolution is the number of points on a segment of length 1.

Log the configuration: `LogCfg( );`

Writes into ulam_cfg.txt number of loop steps, deltaT, 1/h, lambda, ca, nb. of blocks, nb. of fixed blocks, nb. of unknowns.

Prepare or erase a log file: `LogPrepare(File Id, `description`);`

The description is optional.

Log the real time: `LogRuntime(File Id);`

Prepare a sensor file: `SensorEnable(File Id);`

Sensor files are called sensor*Id*.txt

Log sensor data: `SensorWrite(File Id, x position, y position);`

Save non dimensional time T, x, y position, x,y velocities, pressure and domain id.

Move interfaces with solution: `AdvanceTimestep(delta t, 1);`

Bold designates fixed names in Ulambator, *Italic* user defined names and Roman are variables, with minuscules double, staring with a majuscule integer and in 'parenthesis' char arrays.