What is Zebraflow?

Zebraflow is a cell and plasma flow numerical solver written in C programming language. In the posited example, blood flow in a zebrafish trunk network (obtained from confocal fluorescence microscopy imaging on a wild-type (WT) zebrafish embryo at 54 hours post fertilization) is analyzed using a multiphase simulation model with deformable red blood cells (RBCs/erythrocytes) embedded in plasma:

1) Fluid phase momentum and pressure is solved using the Lattice Boltzmann Method (LBM);

2) RBC deformation forces are calculated using a coarse-grained spectrin model (CGDSM) of the RBC membrane shell;

3) Momentum exchange between the plasma and RBCs are solved with the Immersed boundary method (IBM).

The LBM, CGSM and IBM solvers are organized into a single source code called “ZebraflowWT.c”

Repository organization

I have organized the folder structures for the convenience of the user’s first usage.

Download the sourcecodes into your working folder. Next, make a subfolder in the working folder named InputData. Download the compressed input files ending with extensions .gz into /InputData. Uncompress them in a linux terminal using the command:

gunzip *.gz

Download the .tar.gz archives into the working folder and extract them using the commands:

tar xzf *.tar.gz

After extraction into your working folder, please do not rename the folders nor move the files out of the specific folders as this will break the codes’ execution due to incorrect pathname defined in the source code. If you do insist on reorganization then please change the pathname in the source codes accordingly. The working folder organization should look like this:

Name	Size	Type	Modified
CGSMDataStructure	—	Folder	Yesterday
Documentation	—	Folder	01:33
InputData	—	Folder	Yesterday
LBMGridDataStructure	—	Folder	Yesterday
PostProcessedData	—	Folder	Yesterday
SimulationOutputData	—	Folder	Yesterday
WallMeshDataStructure	—	Folder	Yesterday
README.pdf	372.9 kB	Document	01:31
OMPenvironment.txt	111 bytes	Text	Yesterday
PostProCelltoVTK4.c	11.2 kB	Text	Yesterday
PostProWSSandPressure.c	10.8 kB	Text	Yesterday
ZebraflowWT.c	426.3 kB	Text	Yesterday

There are 3 source codes for compilation and execution in your working folder:

1) **ZebraflowWT.c** – this is the zebrafish vascular network flow solver using LBM, CGSM and IBM. Auxiliary data structure generated by the code is outputted into /CGSMDataStructure, /LBMGridDataStructure and /WallMeshDataStructure. Flow results generated are outputted into /SimulationOutputData.
2) **PostproCelltoVTK4.c** – this is a code for generating triangulated RBC meshes from the ".csv" files located in /SimulationOutputData. Outputs from this code are deposited in /PostProcessedData.
3) **PostProWSSandPressure.c** – this is a code for generating wall shear stress and pressure distribution maps on the zebrafish vascular network from the ".csv" files located in /SimulationOutputData. Outputs from this code are deposited in /PostProcessedData.

Folders and their contents:

1. /InputData

1.1. ZFISHRBCNODES.dat – the resting (zero-strain) configuration of RBC membrane nodes
1.2. ZFISHRBCTRIANGLES.dat – the resting (zero-strain) configuration of RBC membrane triangles
1.3. NucleusEdges.dat – the edge-element list for two nodes in each nucleus edge filament
1.4. NucleusNodes.dat – the resting (zero-strain) configuration of RBC nucleus nodes
1.5. AllCellMembStart.aa & *.ab (2 files) – the starting configuration for RBC membranes in the simulation domain *note that this is not the zero-strain state
1.6. AllCellNucStart.csv – the starting configuration for RBC nuclei in the simulation domain *note that this is not the zero-strain state
1.7. FLUIDdomainBindex.csv – the simulation grid definition file that specifies lumen grid nodes, lumen wall grid nodes and pressure boundary grid nodes on the main LBM grid.
1.8. DAparentBoundary.csv – the simulation grid definition file that specifies boundary grid nodes on the periodic LBM grid segment feeding RBCs into the artery blood flow
1.9. PCV1parentBoundary.csv – the simulation grid definition file that specifies boundary grid nodes on the periodic LBM grid segment feeding RBCs into the larger vein
1.10. PCV2parentBoundary.csv – the simulation grid definition file that specifies boundary grid nodes on the periodic LBM grid segment feeding RBCs into the smaller vein
1.11. FLUIDdomainStart.aa to *.aj (10 files) – these are the files containing the pressure and velocity values on the main LBM grid nodes for initializing the simulation.
1.12. DAparentdomainStart.csv – the pressure and velocity values on the LBM grid nodes for initializing the simulation in the periodic segment feeding RBCs into the artery blood flow
1.13. PCV1parentdomainStart.csv – the pressure and velocity values on the LBM grid nodes for initializing the simulation in the periodic segment feeding RBCs into the larger vein
1.14. PCV2parentdomainStart.csv – the pressure and velocity values on the LBM grid nodes for initializing the simulation in the periodic segment feeding RBCs into the smaller vein

1.15. *WallMesh* files (14 files) – the coordinates, data-structures and normal vectors of the lumen wall mesh in the main grid and in the 3 periodic grids.

The other folders are empty upon extraction. You have to run the code to populate them.

2. /CGSMDataStructure

2.1. MembEquilibriumData.dat – the coarse-grained spectrin parameters set based on the defined resting membrane shear modulus

2.2. MembTriAreaEq.dat – the local triangle element area on the RBC membrane mesh under zero strain

2.3. NucEquilibriumData.dat – the coarse-grained filament parameters, set to be the same as RBC membrane spectrin

2.4. RBC_edgeNODES_TRIANGLES_list.dat – the list of node and triangle elements connected to the specific edge elements

2.5. RBC_NODEtoNODE_list.dat – the local connectivity list of neighboring membrane nodes around a specific membrane node

2.6. RBC_TRIANGLEtoNODE_list.dat – the local connectivity list of neighboring membrane triangle elements around a specific membrane node

3. /LBMGridDataStructure

3.1. DOMID.csv – the cartesian domain outputted for visualizing the 1-D array IDs used for OpenMP threading of the 3-D LBM main grid

3.2. DOMID_1.csv, DOMID_2.csv, DOMID_3.csv – the cartesian domains for visualizing 1-D array IDs used for OpenMP threading of the 3 periodic LBM grids.

3.3. DA.. PCVparent1.. PCVparent2Domain.csv – the cartesian domains for visualizing lumen grid points, lumen wall grid points and periodic boundary grid points in the 3 periodic LBM grids

3.4. LumenBlock.vtk – the full 3D cartesian block domain of the main LBM grid showing repulsion force fields in the extraluminal (non-lumen) grid nodes, used for preventing RBCs penetrating into non lumen space.

4. /WallMeshDataStructure

4.1. Connectivity files for the lumen wall mesh are found here.

5. /SimulationOutputData

5.1. AllCellMemb-XXXXXXXXXX.csv – the RBC membrane node coordinates, velocity and internal forces data at each output (XXXXXXXXX) time frame.

5.2. AllCellNucleus-XXXXXXXXXX.csv – the RBC nucleus node coordinates, velocity and internal forces data at each output (XXXXXXXXX) time frame.

5.3. FLUIDdomain-XXXXXXXXX.csv – the values for pressure, velocity and body forces in the main LBM grid at each output (XXXXXXXXX) time frame.

5.4. DA.. PCV1.. PCV2.. parentdomain-XXXXXXXXX.csv – the values for pressure, velocity and body forces in the 3 periodic LBM grids at each output (XXXXXXXXX) time frame.

6. /PostProcessedData

6.1. AllCellMemb-XXXXXXXXX.vtk – The mesh file for RBC membrane surfaces at each output (XXXXXXXXX) time frame for analysis in Paraview

6.2. Wallpressure-XXXXXXXXXX.vtk – The mesh file for zebrafish trunk vascular network at each output (XXXXXXXXX) time frame for analysis in Paraview

7. /Documentation

7.1. ZebraflowDocumentation.pdf – this file that you are reading.

7.2. 1CardiacCycle.mp4 – a movie file of 1 cardiac cycle of simulation using the source codes provided.

Compiling

I recommend GCC compiler for stability. I did not test this code compilation with other compilers. I recommend Geany as a compatible progamming text editor GUI.
As this code is optimized for OpenMP parallelization, please build the executable using the following options.

1) If compiled via shell/terminal:
   gcc -fopenmp -O2 -Wall -mcmodel=large -o ZebraflowWT ZebraflowWT.c -lm

2) if compiled in geany, open ZebraflowWT.c in geany and set the following build command:
   gcc -fopenmp -O2 -Wall -mcmodel=large -o “%e” “%f” -lm

Running

I recommend all compilation and execution in Linux or Unix operating systems. This code has been tested successfully on Centos6, 7 and 8, Manjaro 21.1, 21.2, Linux Mint 20. However, I expect most Linux OS generally to be compatible.

Open a terminal or shell for text-line command in your preferred Linux OS. Before running, please set the environment variables for OpenMP parallelization in the terminal/shell. An example of the variable setting can be found in the file “OMPenvironment.txt” in the base folder:

```
export OMP_PROC_BIND=”FALSE”
export OMP_STACKSIZE=”4000000”
export OMP_NUM_THREADS=60
export OMP_DYNAMIC=”FALSE”
```

To check that this is done correctly type the command “env”. For my work I used an AMD Ryzen Threadripper 3970x 32-core processor with 64 threads. Here I have assigned 60 threads for the running of the code; based on your system, set the appropriate number of threads (ie. export OMP_NUM_THREADS=depends_on_your_system).

Now you can run the simulation in the terminal/shell.

1) To run within the terminal/shell:
   ./ZebraflowWT

2) To run in the background (simulation update script outputted in file “nohup.out”):
   nohup ./ZebraflowWT &

Analysis

Compile and run PostProCelltoVTK4.c and PostProWSSandPressure in the same manner as you did with ZebraflowWT.c.

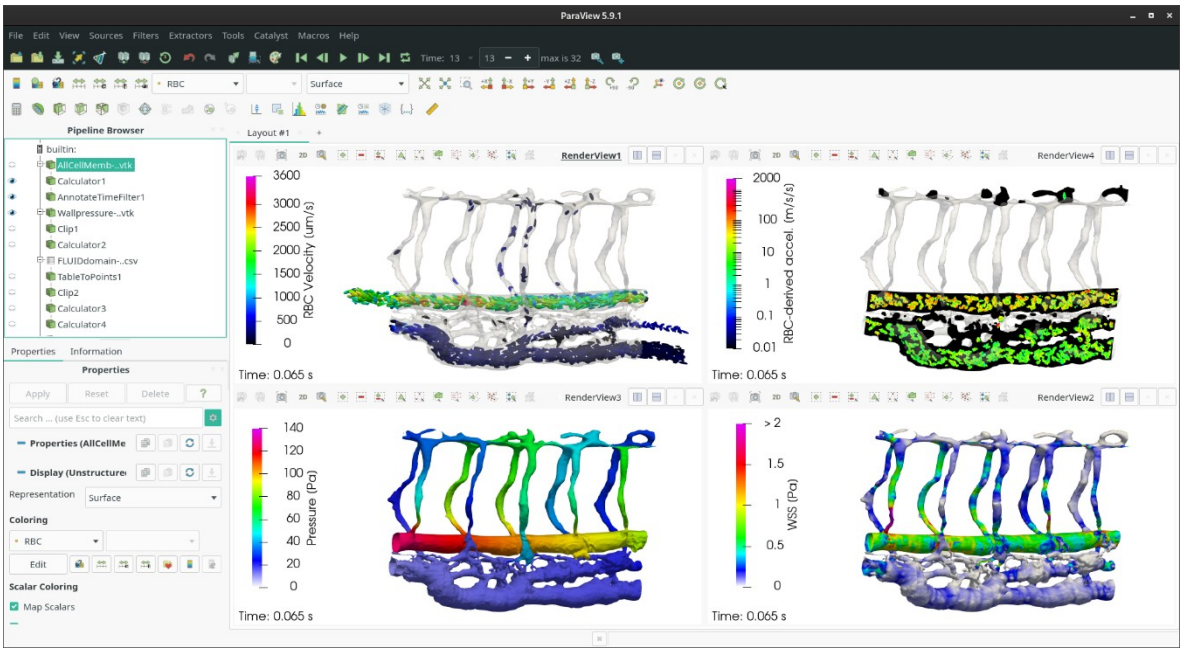
Upon execution of the two PostPro scripts, two sets of mesh data will be generated: AllCellMemb-XXXXXXXXX.vtk and Wallpressure-XXXXXXXXXX.vtk. Open these files using ParaView - Open-source, multi-platform data analysis and visualization application

An example of the analysis environment in Paraview has been prepared as a walkthrough.

Start the Paraview program. Open the file tab in Paraview, click on Load State and select the file “1CycleExample.pvsm” located in the /PostProcessedData folder.

The Load State Options will give several options, select the Choose File Names option and set the correct pathways to 3 groups of files: 1) AllCellMemb-XXXXXXXXX.vtk and 2) Wallpressure-XXXXXXXXXX.vtk are located in the /PostProcessedData folder and 3) FLUIDdomain-XXXXXXXXXX.csv files are located in the /SimulationOutputData folder.

You will now see the files loaded in Paraview looking something like this:



Paraview offers data analysis features that are well documented. This example environment has provided the basic setup required to reproduce results discussed in my paper. I encourage you to explore its features and perform the appropriate data segmentation and analyses from the simulation generated data. You can also contact me or future versions of the repository for additional scripts with regards to data segmentation and analysis without using Paraview.