CompBioMed
A Centre of Excellence in Computational Biomedicine

Webinar series

Webinar #10
HemeLB - Simulation of cardiovascular flow on high performance computers

9 September 2019

The webinar will start at 1pm CEST

Presenter:
Dr Jon McCullough (UCL)

Moderator:
Ben Czaja (UvA)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 675451

The webinar series is run in collaboration with:

VPH Institute
Building the Virtual Physiological Human

Webinar #10
HemeLB - Simulation of cardiovascular flow on high performance computers

9 September 2019

Welcome!

Presenter:
Dr Jon McCullough (UCL)

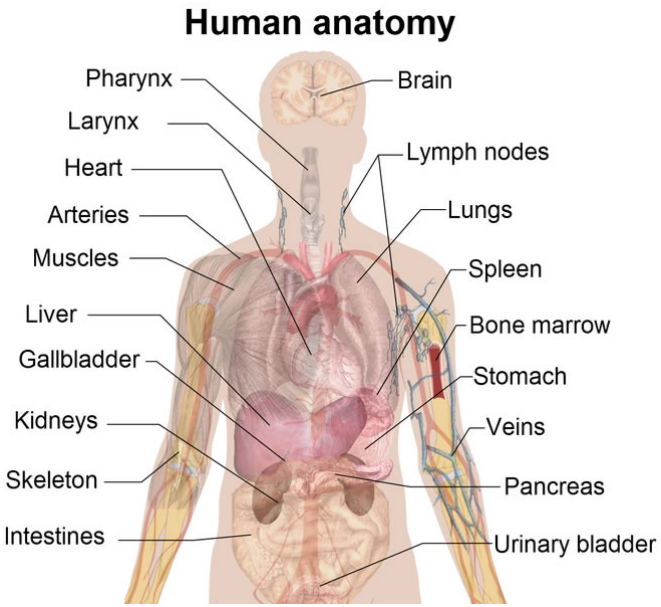
Moderator:
Ben Czaja (UvA)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 675451

The webinar series is run in collaboration with:

VPH Institute
Building the Virtual Physiological Human

The Challenge of Medicine



Some of them anyway:

- Human body is comprised of multiple complex sub-systems
- Various factors further individualise each body - age, gender, weight, lifestyle, genetics, . . .
- Many medical treatments are invasive and require all these to be considered on a case-by-case basis

Question: How can outcomes for patients be improved?

Patient Specific Modelling

Answer: Assist clinicians' ability to understand how a course of treatment will impact a given individual

How: Simulating the patient using a personalised digital replica

Why:

- Multiple options for treatment can be investigated and the optimal course chosen for an individual
- Clinicians can treat a patient with greater confidence
- Digital models can provide non-traditional information about the patient
- Patients can have a clearer understanding of how a treatment will impact them

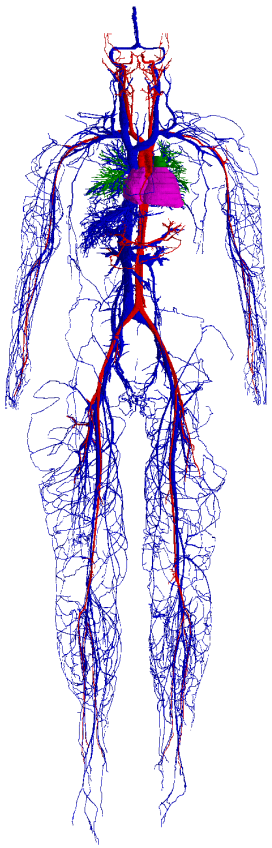
The Virtual Human

Non-invasive scanning technology has been available since 1970s:

- Computed Tomography (CT) scan
  - Images constructed from multiple x-rays
  - Generally better for bones, tumors, chest
- Magnetic Resonance Imaging (MRI)
  - Generally better for soft tissue, esp. brain
  - Images constructed from measuring atomic magnetic interactions

Quality and cost-efficiency is constantly improving with both techniques.

The Virtual Human

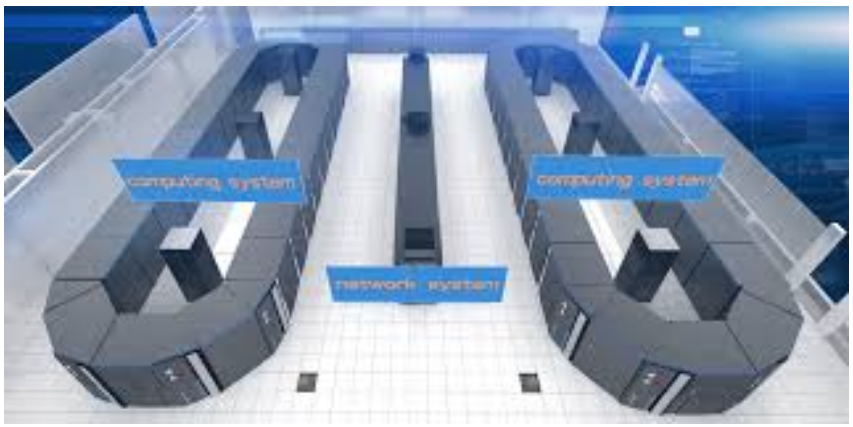


- Develop numerical models to predict behaviour of bodily components - organs, blood, nerves, ...
- Behaviour needs to be captured at multiple length scales
- Codes need to communicate at physiological interfaces
- Resolution needs to be high enough for clinical accuracy
- Ultimately, calculation time needs to be as short as possible

See CompBioMed's Virtual Humans film at - `https://youtu.be/1FvRSJ9W734`

Computers - From Calculators to HPC

- Technological advancements have meant that computers have continually been getting faster[1]
- Largest supercomputer (Sunway TaihuLight, China) currently has over 10.5 million CPU cores
- Performance of top machines $\mathcal{O}(10^{15} - 10^{17})$ floating point operations per second
- Exascale machines ($> 10^{18}$ *flops*) anticpated in early 2020s



[1] Whether this continues is a whole different webinar

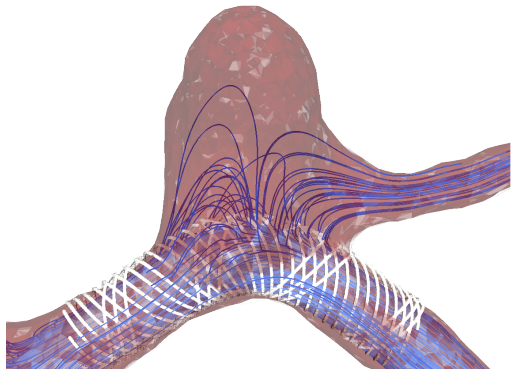
Computers - From Calculators to HPC

- Simulations of full-humans at sufficient resolution currently demand the use of machines on this scale
- Developing codes that perform efficiently on machines of this size is challenging. Concerns include:
  - Communication between cores
  - Load balancing
  - Data visualisation
- Studying models of appreciable size becoming possible on desktop based machines - especially with GPU acceleration (another code issue).

HemeLB - Purpose and Direction

HemeLB is a code developed with the vision of modelling a wide variety of vascular and blood flow problems

Within a virtual (and physical) human, blood flow around the networks of arteries and veins is vital for the communication of information (e.g. oxygen) around the body.



In a virtual human simulation, HemeLB will provide an interconnect between models for other organs - heart, brain, lungs, liver, . . .

HemeLB - History

The initial version of HemeLB was published in 2008, developed within CCS at UCL

Current support for HemeLB is, in particular, provided through - UKCOMES, CompBioMed CoE, QNRF and compute time used on ARCHER (UK), SuperMUC-NG (LRZ, GER), BlueWaters (NCSA, USA), ...

HemeLB - Capabilities and Case Studies

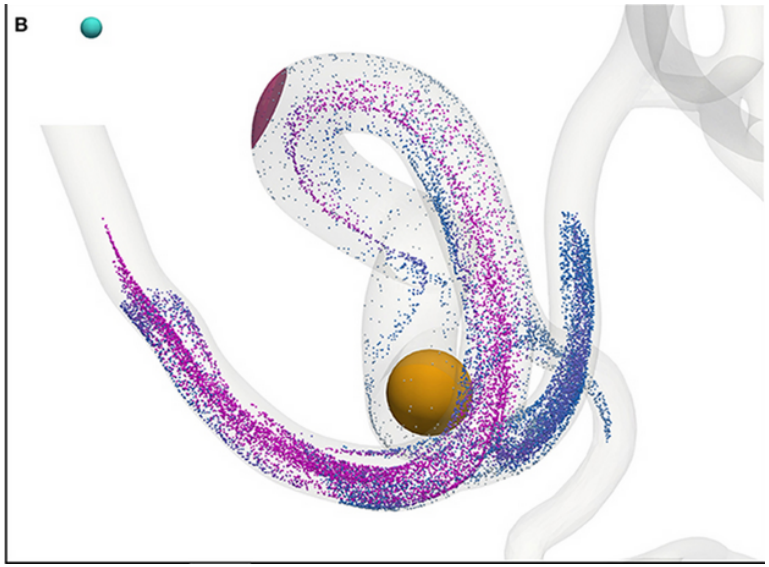
Features of HemeLB:

- C++ code parallelized using standard MPI communications
- CPU code (currently)
- Fluid flow solved using lattice Boltzmann method
- Multiple boundary condition options available
- Optimized for sparse geometries characteristic of vascular geometries
- Code execution designed to scale well on up to hundreds of thousands of cores

HemeLB - Capabilities and Case Studies

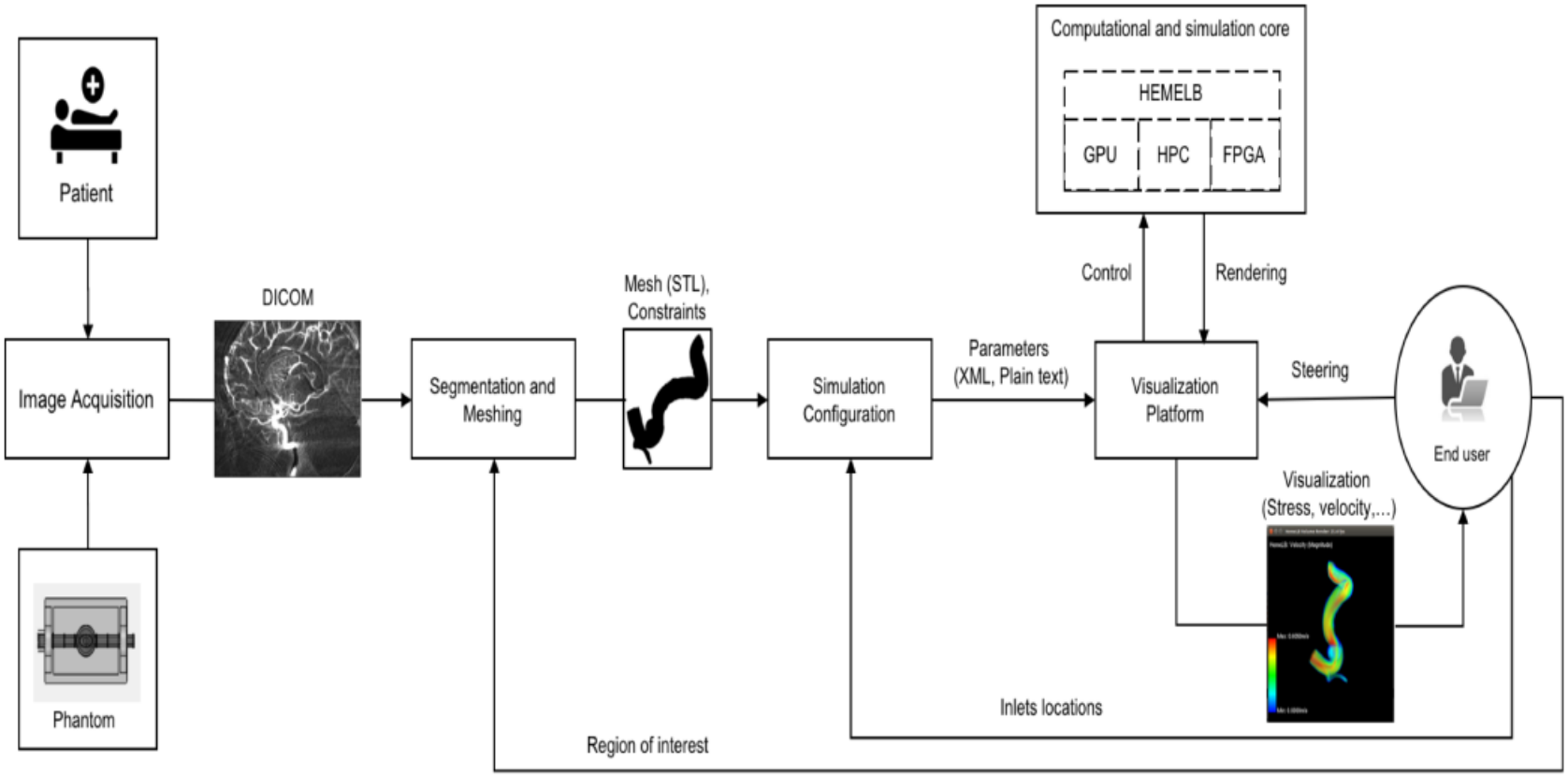
Applications of HemeLB:

- Stent design
- Retinal vascular flow
- Cerebral blood flow
- Magnetic drug targetting



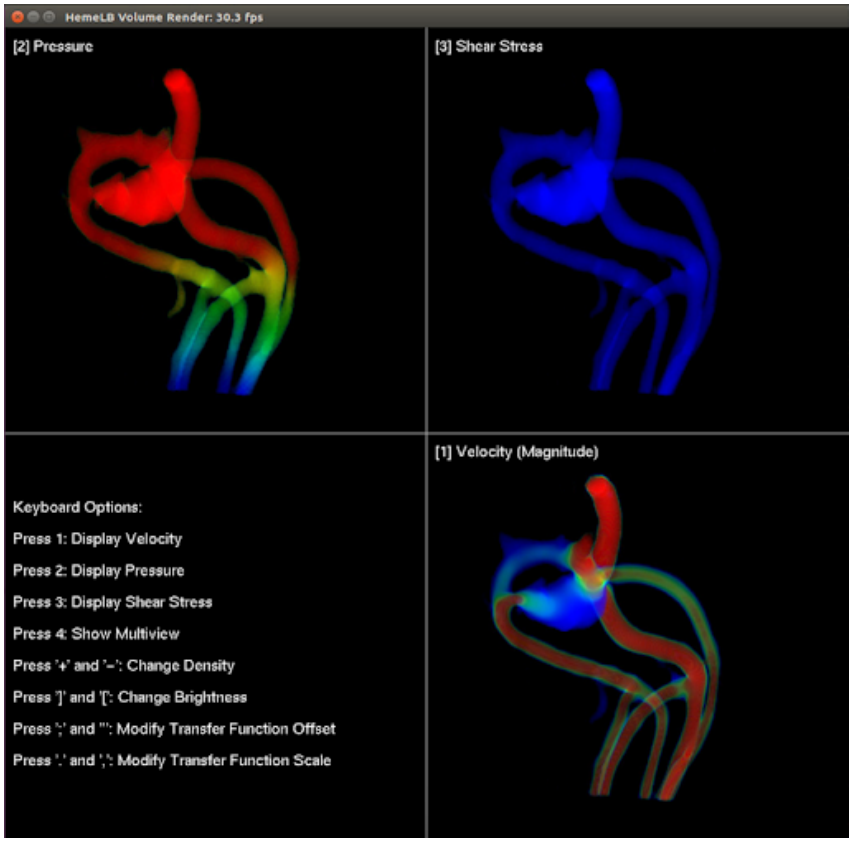
HemeLB - Capabilities and Case Studies

Applications of HemeLB: Aneurysm Pipeline



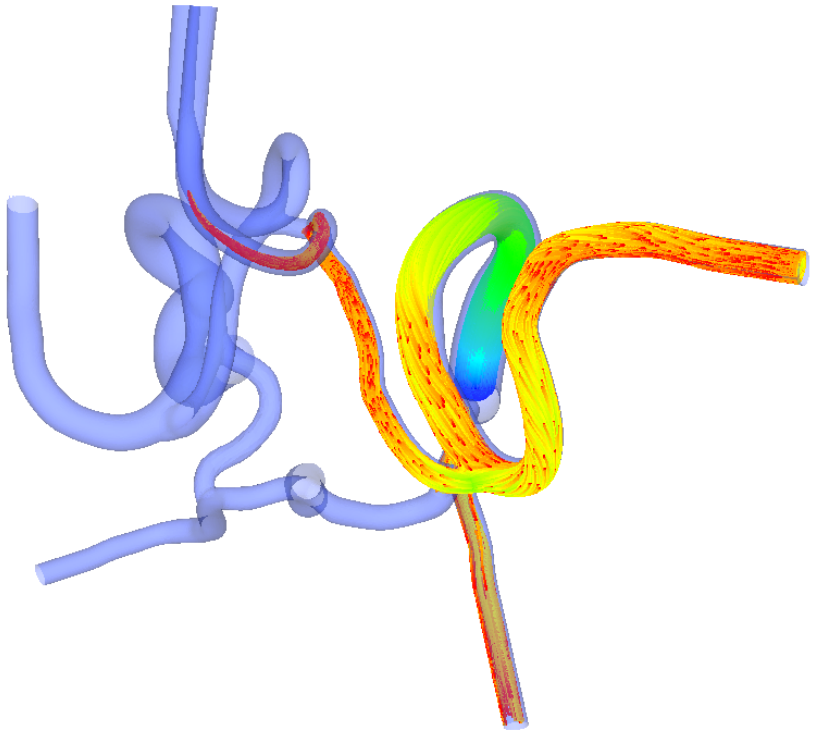
HemeLB - Capabilities and Case Studies

- Real time visualisation, using CUDA
- Current version displays pressure, shear stress and local velocity magnitude
- Frequent discussions with clinicians to see what visualisations make most sense to them



HemeLB - Capabilities and Case Studies

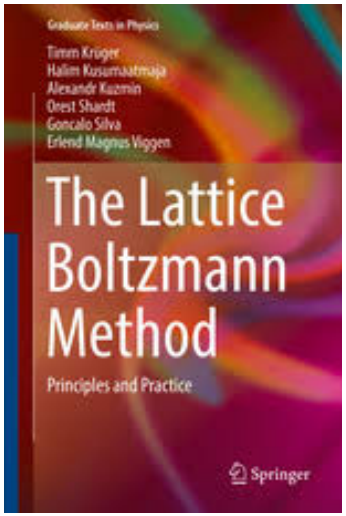
Play video - HemeLB flow in Circle of Willis



HemeLB - Lattice Boltzmann Method

For further information on Lattice Boltzmann Method see CompBioMed Webinar #3 - Lattice Boltzmann method for CompBioMed (incl. Palabos) (Dr Jonas Latt, University of Geneva)

Many textbooks are also available e.g. The Lattice Boltzmann Method: Principles and Practice - Krüger *et al.*

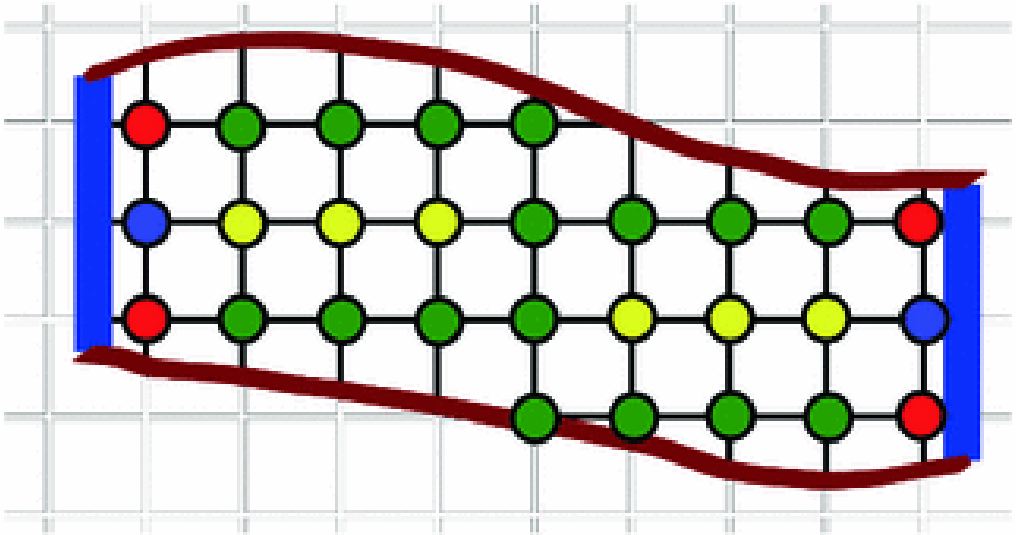


HemeLB - Lattice Boltzmann Method

- Numerical method for solving Boltzmann equation
- Under certain conditions solves Navier-Stokes equations for fluid flow
- Localised algorithm has multiple advantages:
  - Easy to parallelise
  - Straightforward to use on complex boundary shapes
  - Adaptable for advanced flow physics - multiphase, multi-component, non-Newtonian, thermal

HemeLB - Lattice Boltzmann Method

- Determine lattice structure to use - DnQm
  - n = dimensions
  - m = velocity directions
- Discretise domain into nodal locations using a regular cartesian grid
- Identify fluid sites and wall sites



HemeLB - Lattice Boltzmann Method

At each site - assign distributions, $f_i(\mathbf{x}, t)$, in velocity direction i

- Represents the probability of fluid at site $\mathbf{x}$ moving in the i direction at a given point in time (t).
- Often simplified to f_i for clarity
- At equilibrium, these are often determined through an approximation to the Maxwell-Boltzmann distribution

$$f(v) = \left(\frac{m}{2\pi kT}\right)^{3/2} e^{-\frac{mv^2}{2kT}}$$

HemeLB - Lattice Boltzmann Method

Evolution of flow computed by streaming and collision of distributions

1. Collision - HemeLB typically uses the BGK collision function

$$f_i^*(\mathbf{x}, t) = f_i(\mathbf{x}, t) - \frac{\Delta t}{\tau}(f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t))$$

- $f_i(\mathbf{x}, t)$ - pre-collision distribution function
- $f_i^*(\mathbf{x}, t)$ - post-collision distribution function
- τ - BGK relaxation time
- $f_i^{eq}(\mathbf{x}, t)$ - equilibrium distribution function

HemeLB - Lattice Boltzmann Method

Evolution of flow computed by streaming and collision of distributions

2. Streaming

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i^*(\mathbf{x}, t)$$

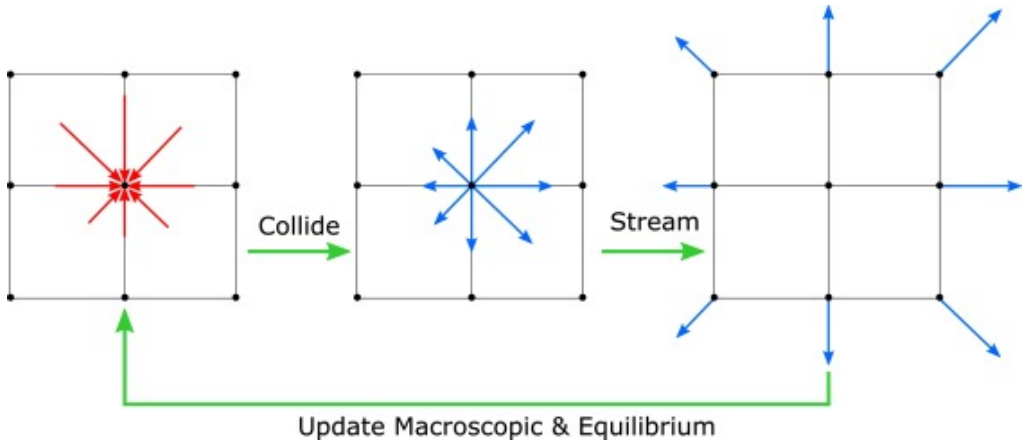


Image: A.R.G Harwood and A.J. Revell, Advances in Engineering Software (2017), Vol. 104, pp. 38-50.

HemeLB - Lattice Boltzmann Method

Local macroscopic properties determined by moments (sum of distributions weighted by velocity direction).

For hydrodynamics:
Zeroth moment: Density $\Rightarrow \rho = \sum_i f_i$
First moment: Momentum $\Rightarrow \rho \mathbf{u} = \sum_i f_i \mathbf{c}_i$
Second moment: Stress tensor $\Rightarrow$

$$S_{\alpha\beta} = -\left(1 - 1/(2\tau)\right) \sum_i \mathbf{c}_{i\alpha} \mathbf{c}_{i\beta} (f_i - f_i^{eq})$$

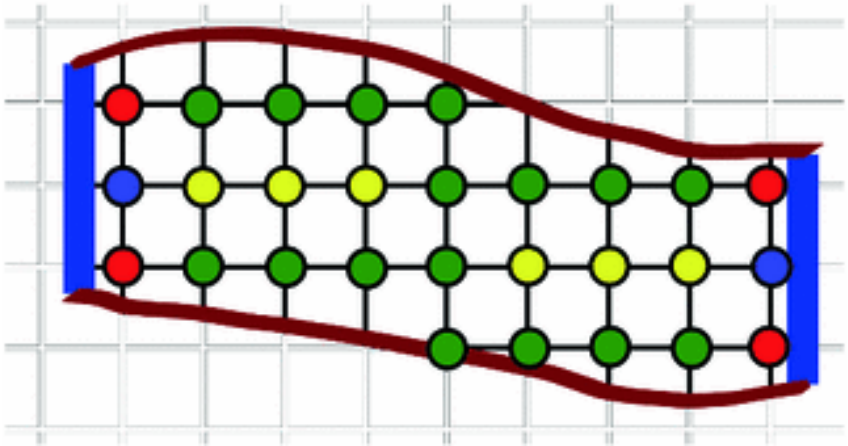
LBM hydrodynamics is only valid in low *Ma* limit, a result of this is:

$$\text{Viscosity} \Rightarrow \nu = \frac{1}{3}\left(\tau - \frac{1}{2}\right)\frac{(\Delta x)^2}{\Delta t}$$

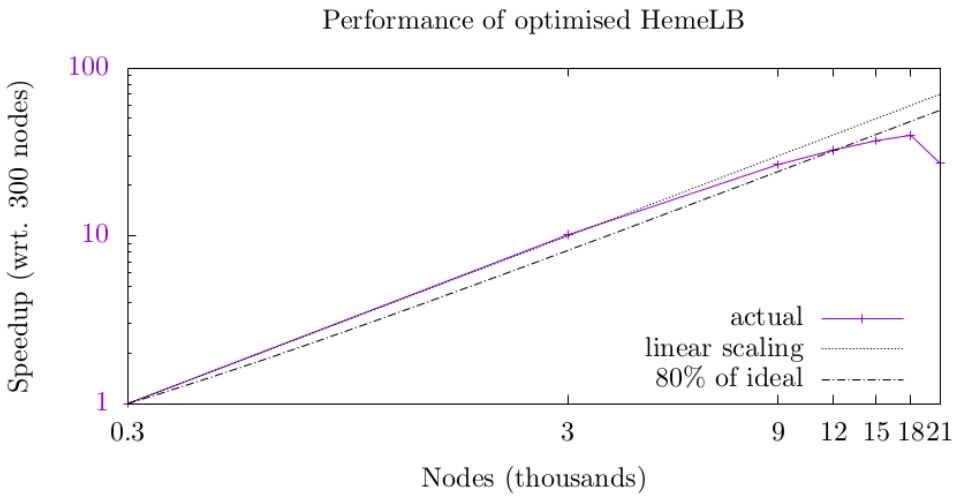
HemeLB - Lattice Boltzmann Method

Boundary Conditions:

- Population values streaming in from outside the domain are unknown
- Construct unknown values based on desired boundary behaviour
- HemeLB allows for multiple options including solid walls and velocity or pressure based in/outlets



HemeLB - Scaling Performance on HPC



Non-ideal performance seen due to:

- Load distribution between processors
- Reliability/performance of processors
- Frequency of I/O operations
- Communication between groups of nodes
- Architecture of HPC facility

HemeLB - Running a Simulation

HemeLB is designed to operate on large-scale high performance computers - university/regional/national clusters

Still able to be run on laptops with any unix-based environment

- Linux/Mac - native
- Windows 10 - WSL (Windows Subsystem for Linux)
- Windows 7 - Sorry!

Does require familiarity of command line interface

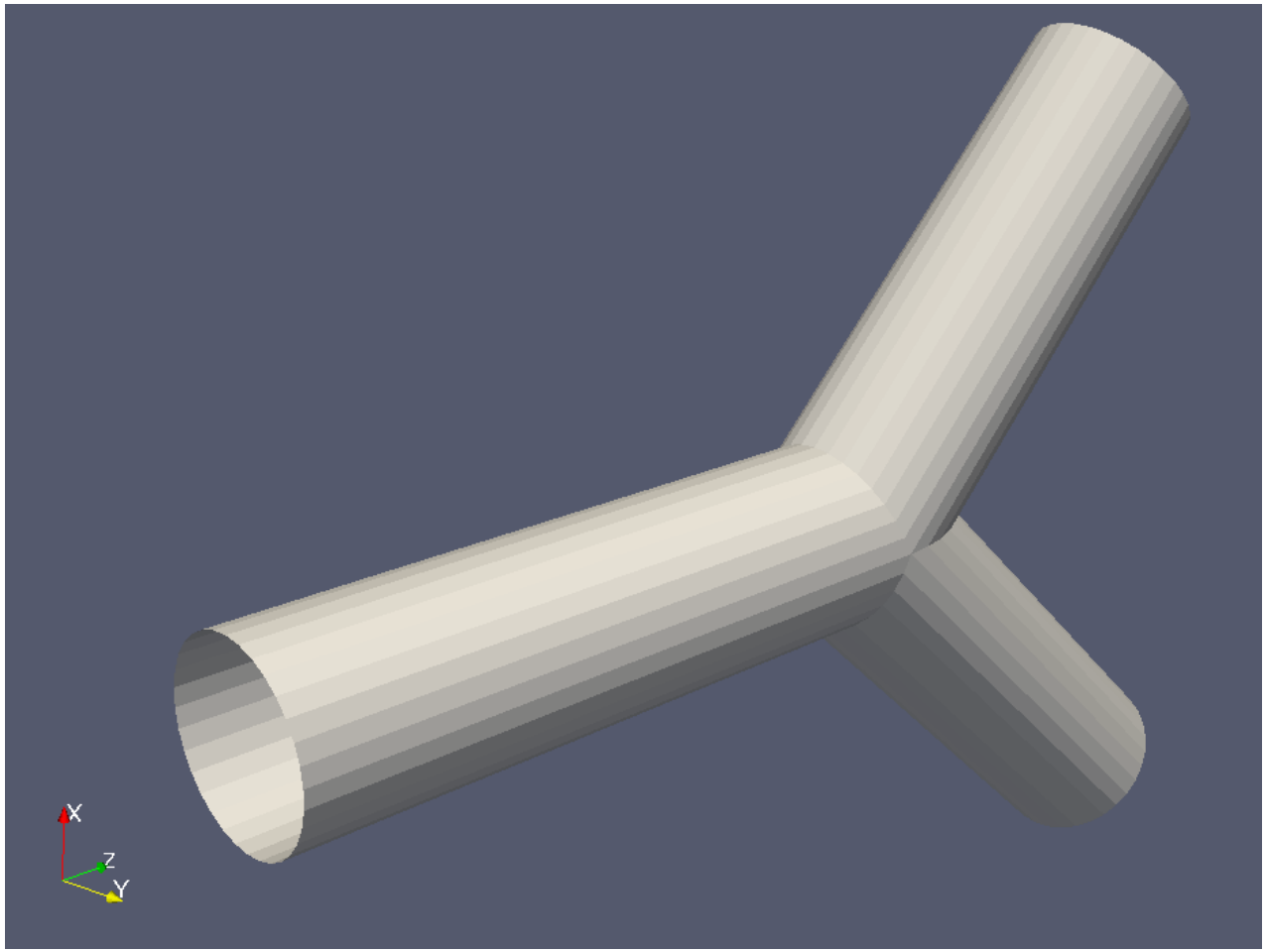
HemeLB - Running a Simulation

Installation and compilation:

- Tools needed for compilation: CMake, MPI (e.g. OpenMPI), GCC
- Other dependencies installed as part of initial compilation process - see online/`README` instructions
- LBM parameters configured into compilation flags - lattice choice, collision kernel, wall and in/outlet boundary conditions
- Download HemeLB source code from `www.hemelb.org`
- Install and compile code as per instructions
- Test case in examples - see website tutorials

HemeLB - Running a Simulation

Flow due to a pressure gradient through a bifurcation



HemeLB - Running a Simulation

```
<?xml version="1.0"?>
<hemelbsettings version="3">
  <simulation>
    <step_length units="s" value="1e-5"/>
    <steps units="lattice" value="5000"/>
    <stresstype value="1"/>
    <voxel_size units="m" value="5e-5"/>
    <origin units="m" value="(0.0,0.0,0.0)"/>
  </simulation>
  <geometry>
    <datafile path="bifurcation.gmy"/>
  </geometry>
  <initialconditions>
    <pressure>
      <uniform units="mmHg" value="0.001"/>
    </pressure>
  </initialconditions>
  <monitoring>
    <incompressibility/>
  </monitoring>
```

Lattice Boltzmann parameters

Geometry file

Initial conditions

HemeLB - Running a Simulation

```
<inlets>
  <inlet>
    <condition subtype="cosine" type="pressure">
      <amplitude units="mmHg" value="0.0"/>
      <mean units="mmHg" value="0.001"/>
      <phase units="rad" value="0.0"/>
      <period units="s" value="1"/>
    </condition>
    <normal units="dimensionless" value="(4.15565e-13,1.44689e-12,1)"/>
    <position units="lattice" value="(165.499,35.8291,3)"/>
  </inlet>
</inlets>
<outlets>
  <outlet>
    <condition subtype="cosine" type="pressure">
      <amplitude units="mmHg" value="0.0"/>
      <mean units="mmHg" value="0.0"/>
      <phase units="rad" value="0.0"/>
      <period units="s" value="1"/>
    </condition>
    <normal units="dimensionless" value="(0.707107,-2.4708e-12,-0.707107)"/>
    <position units="lattice" value="(26.2137,35.8291,372.094)"/>
  </outlet>
  <outlet>
    <condition subtype="cosine" type="pressure">
      <amplitude units="mmHg" value="0.0"/>
      <mean units="mmHg" value="0.0"/>
      <phase units="rad" value="0.0"/>
      <period units="s" value="1"/>
    </condition>
    <normal units="dimensionless" value="(-0.707107,-4.27332e-12,-0.707107)"/>
    <position units="lattice" value="(304.784,35.8291,372.094)"/>
  </outlet>
</outlets>
```

Inlet boundary conditions

Outlet boundary conditions

HemeLB - Running a Simulation

```
  <properties>
    <propertyoutput file="whole.dat" period="100">
      <geometry type="whole" />
      <field type="velocity" />
      <field type="pressure" />
    </propertyoutput>
    <propertyoutput file="inlet.dat" period="100">
      <geometry type="inlet" />
      <field type="velocity" />
      <field type="pressure" />
    </propertyoutput>
  </properties>
</hemelbsettings>
```

Data output information

HemeLB - Running a Simulation

Execute with:

```
mpirun -n N <hemelb executable address> -in <input
file *.xml address> -out <output directory address>
```

```
[Rank 0000000, 6.799901e-01 s, 00000000055328 kB] :: INITIALISE
[Rank 0000000, 6.801260e-01 s, 00000000055328 kB] :: ----------
[Rank 0000000, 6.802080e-01 s, 00000000055328 kB] :: --> loading input and decomposing geometry
[Rank 0000000, 7.014470e-01 s, 00000000060904 kB] :: ----> opened data file ./bifurcation.gmy
[Rank 0000000, 7.015882e-01 s, 00000000060904 kB] :: ----> reading preamble
[Rank 0000000, 7.017149e-01 s, 00000000060904 kB] :: ----> reading header (start)
[Rank 0000000, 7.042514e-01 s, 00000000066640 kB] :: ----> reading header (end)
[Rank 0000000, 7.049118e-01 s, 00000000066636 kB] :: ----> non-empty blocks: 5330
[Rank 0000000, 7.049396e-01 s, 00000000066636 kB] ::           total blocks: 18450
[Rank 0000000, 7.050843e-01 s, 00000000066636 kB] ::                  ratio: 0.288889
[Rank 0000000, 7.051701e-01 s, 00000000066636 kB] ::                  sites: 2010048
[Rank 0000000, 7.052528e-01 s, 00000000066636 kB] :: ----> blockInformation.size(): 5330
[Rank 0000000, 7.053350e-01 s, 00000000066636 kB] ::  fluidSitesOnEachBlock.size(): 0
[Rank 0000000, 7.054173e-01 s, 00000000066636 kB] ::           blockWeights.size(): 0
[Rank 0000000, 7.054991e-01 s, 00000000066636 kB] :: ----> is blockInformation.size() == nonEmptyBlocks? yes
[Rank 0000000, 7.081596e-01 s, 00000000066636 kB] :: ----> not optimising decomposition
[Rank 0000001, 7.090890e-01 s, 00000000017260 kB] :: ----> load distribution: 0.000281
[Rank 0000000, 7.082588e-01 s, 00000000066636 kB] :: ----> basic decomposition (start)
[Rank 0000000, 7.095788e-01 s, 00000000067592 kB] :: ----> basic decomposition (end)
[Rank 0000000, 7.096923e-01 s, 00000000067592 kB] :: ----> read blocks (start)
[Rank 0000001, 7.282304e-01 s, 00000000017784 kB] :: ----> blockInformation.size(): 5330
[Rank 0000002, 7.281983e-01 s, 00000000017412 kB] :: ----> blockInformation.size(): 5330
[Rank 0000003, 7.282003e-01 s, 00000000016404 kB] :: ----> blockInformation.size(): 1931
[Rank 0000000, 2.023984e+00 s, 00000000694832 kB] :: ----> read blocks (end)
[Rank 0000000, 2.121580e+00 s, 00000000695148 kB] :: --> lattice data
[Rank 0000000, 3.151832e+00 s, 00000002014288 kB] :: ----> gathering lattice information (start)
[Rank 0000000, 3.151998e+00 s, 00000002014288 kB] :: ----> gathering lattice information (end)
[Rank 0000000, 4.372778e+00 s, 00000002312332 kB] :: --> neighbouring data manager
[Rank 0000000, 4.373169e+00 s, 00000002312332 kB] :: --> lattice-Boltzmann model
[Rank 0000000, 4.620966e+00 s, 00000002672220 kB] :: ------------------
[Rank 0000000, 4.621000e+00 s, 00000002672220 kB] :: INITIALISE FINISHED
[Rank 0000000, 4.663895e+00 s, 00000002672220 kB] :: SIMULATION STARTING
[Rank 0000000, 4.663959e+00 s, 00000002672220 kB] :: ------------------
[Rank 0000000, 2.208945e+01 s, 00000002672660 kB] :: time step 0000200 :: write_image_to_disk 0
[Rank 0000000, 2.208961e+01 s, 00000002672660 kB] :: time step 0000200 :: tau: 0.548000, max_relative_press_diff: 0.000, Ma: 0.000, max_vel_phys: 5.625405e-05
[Rank 0000000, 4.011607e+01 s, 00000002672660 kB] :: time step 0000400 :: write_image_to_disk 0
[Rank 0000000, 4.011613e+01 s, 00000002672660 kB] :: time step 0000400 :: tau: 0.548000, max_relative_press_diff: 0.000, Ma: 0.000, max_vel_phys: 7.528757e-05
[Rank 0000000, 5.772616e+01 s, 00000002672660 kB] :: time step 0000600 :: write_image_to_disk 0
[Rank 0000000, 5.772624e+01 s, 00000002672660 kB] :: time step 0000600 :: tau: 0.548000, max_relative_press_diff: 0.000, Ma: 0.000, max_vel_phys: 8.581618e-05
[Rank 0000000, 7.520723e+01 s, 00000002672660 kB] :: time step 0000800 :: write_image_to_disk 0
[Rank 0000000, 7.520738e+01 s, 00000002672660 kB] :: time step 0000800 :: tau: 0.548000, max_relative_press_diff: 0.000, Ma: 0.000, max_vel_phys: 1.292713e-04
[Rank 0000000, 9.288469e+01 s, 00000002672660 kB] :: time step 0001000 :: write_image_to_disk 1
[Rank 0000000, 9.288484e+01 s, 00000002672660 kB] :: time step 0001000 :: tau: 0.548000, max_relative_press_diff: 0.000, Ma: 0.000, max_vel_phys: 1.336011e-04
[Rank 0000000, 1.104421e+02 s, 00000002672936 kB] :: time step 0001200 :: write_image_to_disk 0
[Rank 0000000, 1.104422e+02 s, 00000002672936 kB] :: time step 0001200 :: tau: 0.548000, max_relative_press_diff: 0.000, Ma: 0.000, max_vel_phys: 1.336011e-04
[Rank 0000000, 1.283911e+02 s, 00000002673572 kB] :: time step 0001400 :: write_image_to_disk 0
[Rank 0000000, 1.283913e+02 s, 00000002673572 kB] :: time step 0001400 :: tau: 0.548000, max_relative_press_diff: 0.000, Ma: 0.000, max_vel_phys: 1.336011e-04
```

Took approx. 8min on N=4

HemeLB - Visualising a Simulation

The `hemeXtract` tool converts the output.dat file to human-readable format (1). (2) splits this into stepwise files for visualisation in paraview

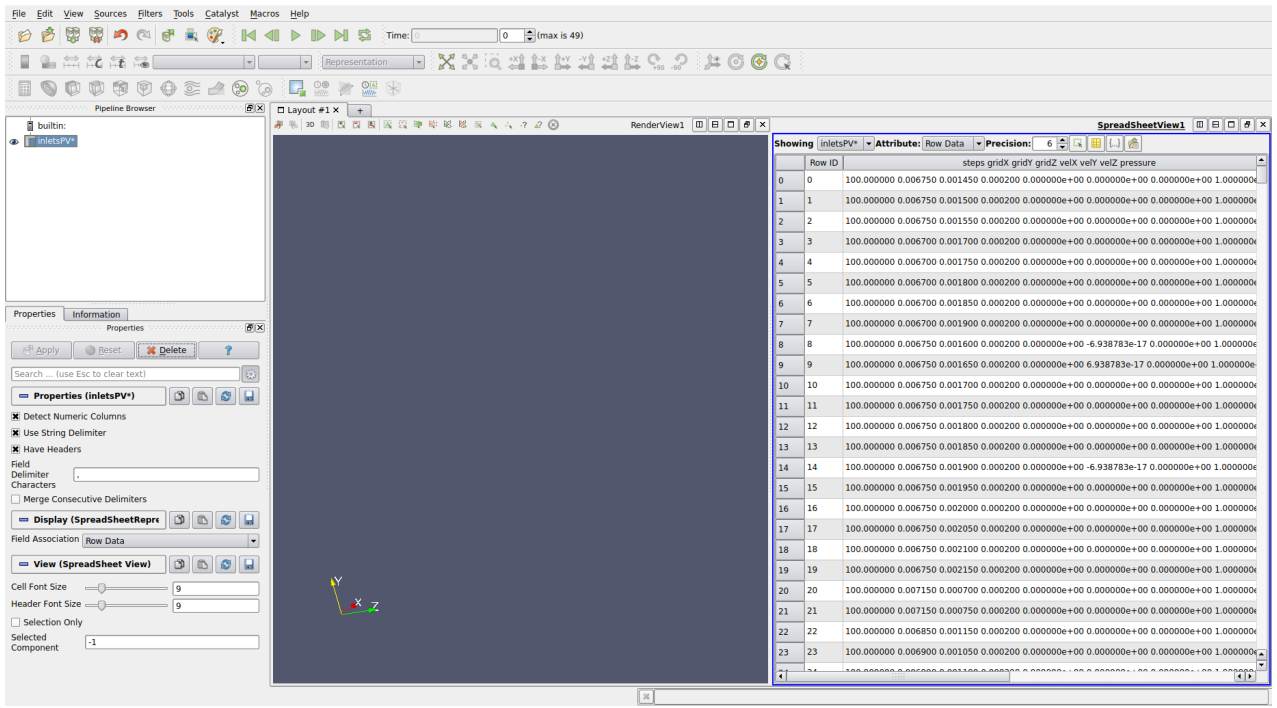
(1) `hemeXtract -X output.dat > readable-output.txt`

(2) `bash paraviewProcessing.sh readable-output.txt paraview-file-name`

Paraview is a widely-used visualisation package - `www.paraview.org/`

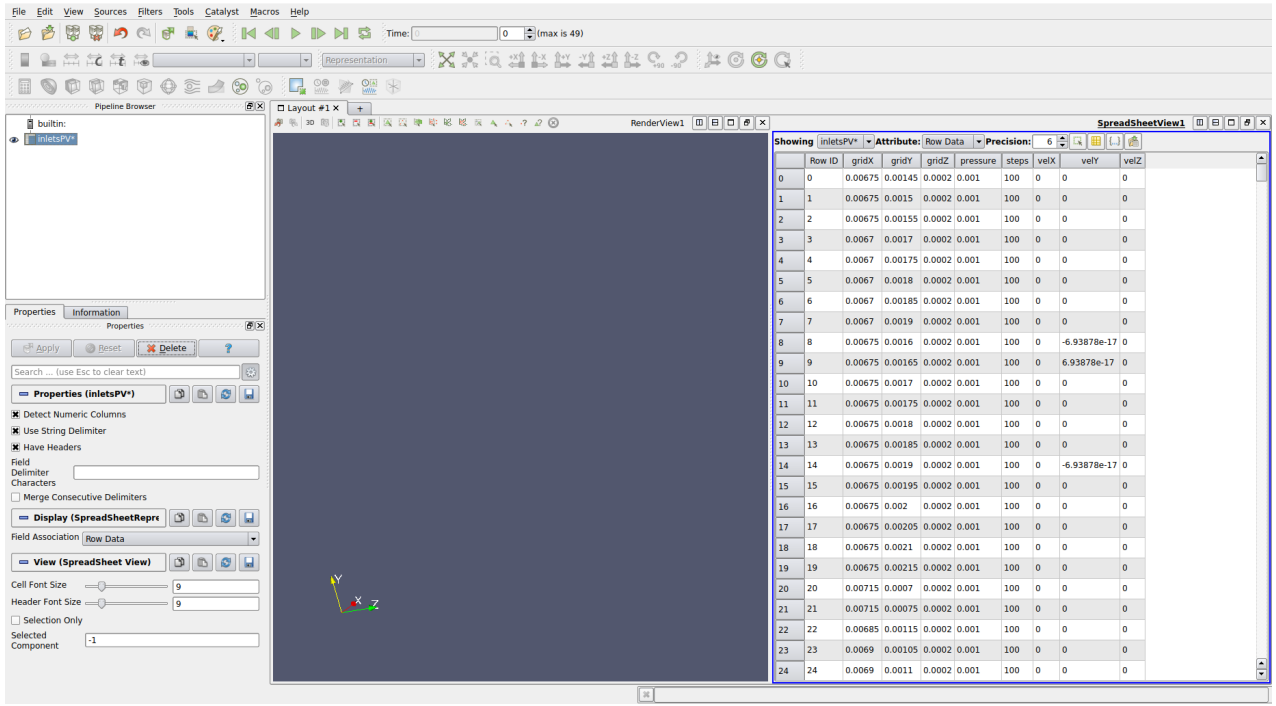
HemeLB - Visualising a Simulation

1. Import paraview-file-nameXX.txt files: File → Open → Navigate to folder → double-click on paraview-file-name..txt (type GROUP)
2. Click green apply button on LHS



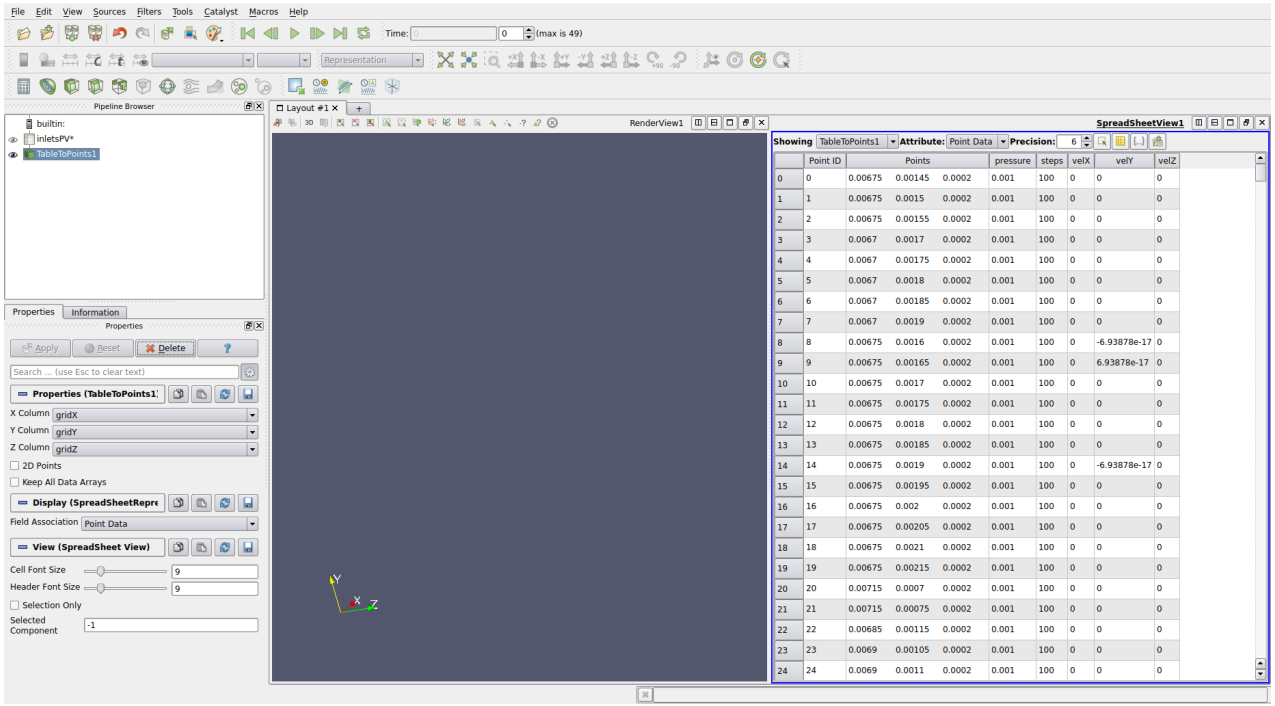
HemeLB - Visualising a Simulation

3. Change default comma (,) in Field Delimeter Characters to a space ( )
4. Click green apply button on LHS (this is a recurring theme in Paraview)



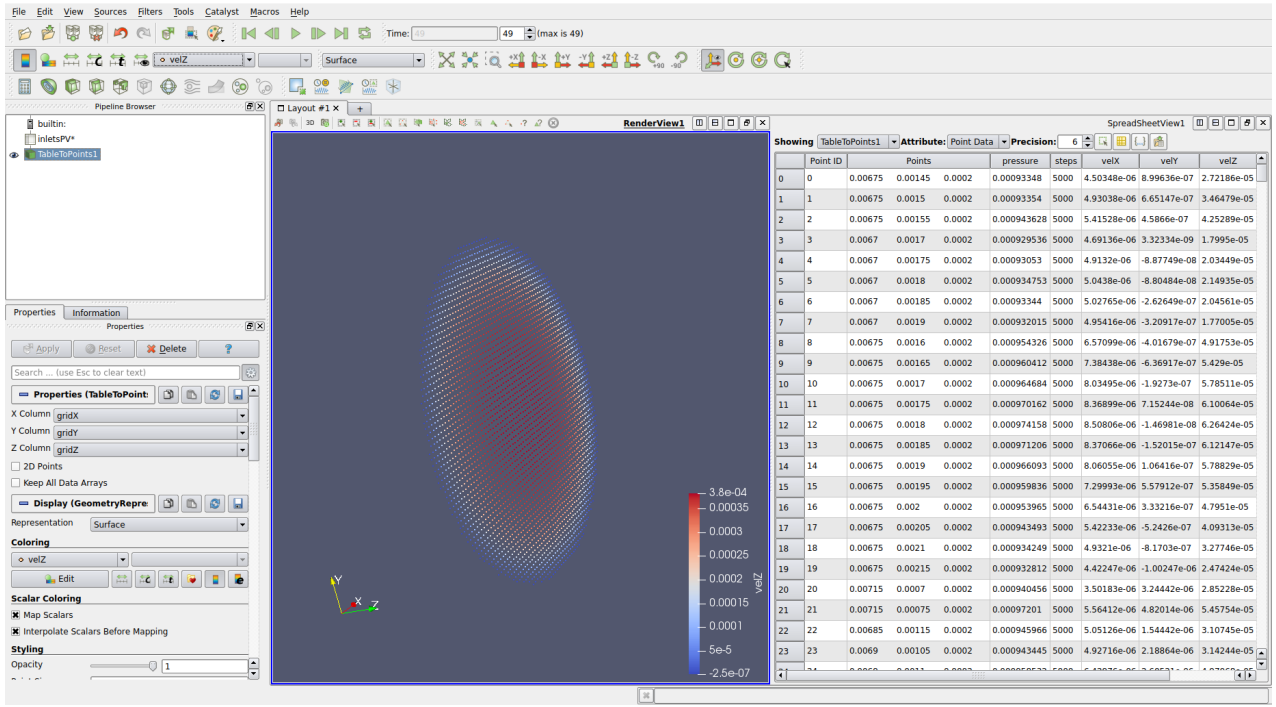
HemeLB - Visualising a Simulation

5. Filters → Alphabetical → Table To Points
6. Change Y Columns to 'gridY' and Z columns to 'gridZ'
7. Click green apply button on LHS



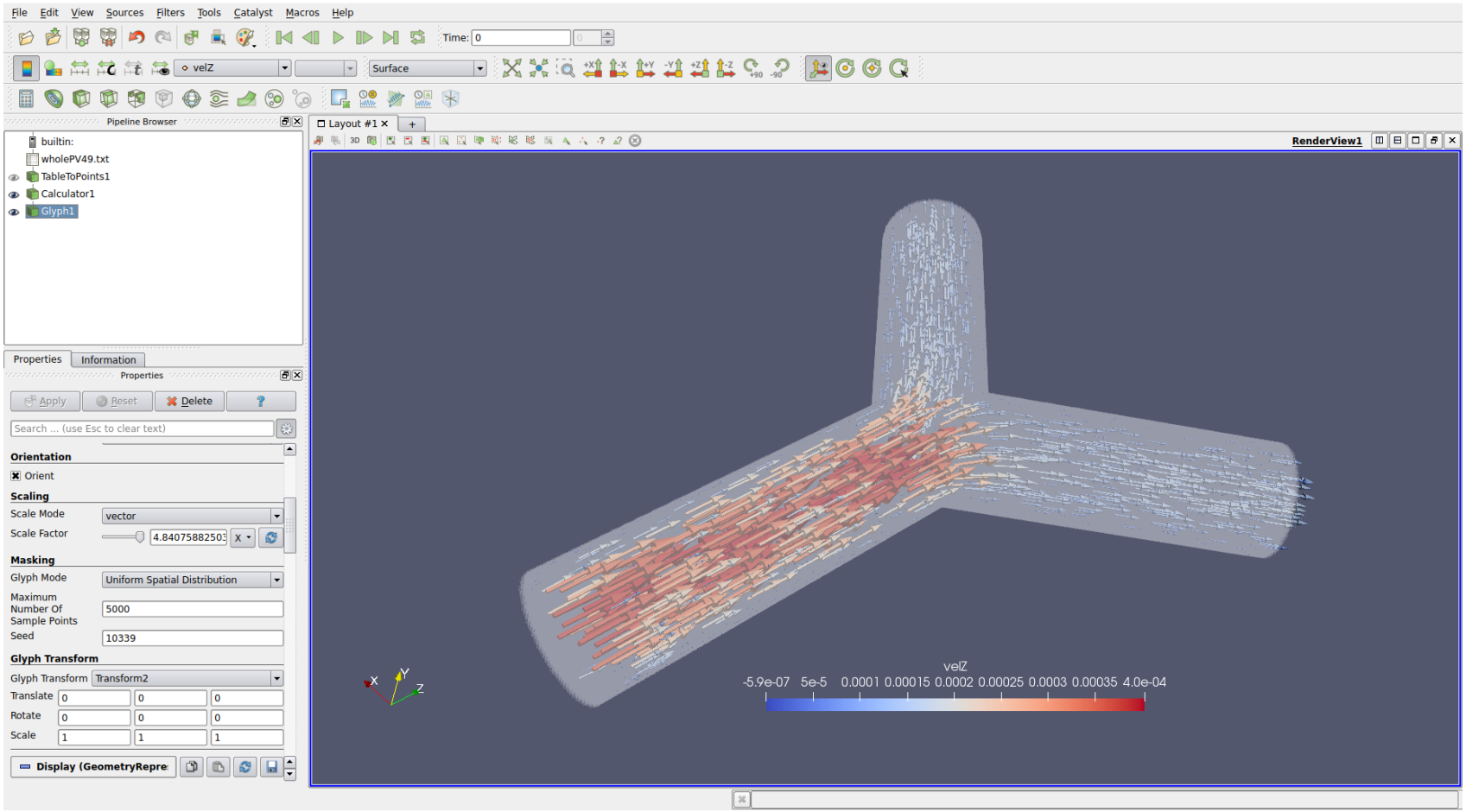
HemeLB - Visualising a Simulation

8. Ensure RenderView panel is active and click on eye next to TableToPoints1 in Pipeline
9. Change rendering option from 'Solid Colour' to 'velZ'
10. Use Play and rescale options to watch time evolution



HemeLB - Visualising a Simulation

Can do plenty more with Paraview - glyph and streamline examples



HemeLB - Future Developments

HemeLB is constantly developing - accuracy and scalability always of concern.

- Coupling to itself and other physiological codes
- Elastic walls - capture the variable shape of vessels
- GPU implementation - enable HemeLB to take advantage of performance advantages of GPU architecture

Summary

HemeLB is numerical code designed for solving blood flows within human-scale vasculatures.

- Optimised for the sparse geometries
- Highly scalable - performance on over 300,000 cores
- Demonstrated capability for simulating highly complicated vasculatures
- Will communicate between multiple codes for simulation of a full virtual human
- Get more information from the HemeLB website - `www.hemelb.org`

Acknowledgements

- Funding through - EPSRC, MRC, STFC, The European Commission, Qatar National Research Fund
  - UKCOMES (EPSRC No. EP/L00030X/1 (2013-2018), EP/R029598/1 (2018-2022))
  - CompBioMed CoE (EU Horizon 2020 (Grant Nos. 675451 and 823712))
- Developers of HemeLB in P.V. Coveney's CCS group at UCL include: M.D. Mazzeo, S. Manos, G.M. Doctors, M.O. Bernabeu, R.W. Nash, D. Groen, H.B. Carver, J. Hetherington, T. Krüger, U.D. Schiller, S. Schmieschek, A. Patronis, R.A. Richardson, J.W.S McCullough

Selected Publications

- M. D. Mazzeo and P. V. Coveney, "HemeLB: A high performance parallel lattice-Boltzmann code for large scale fluid flow in complex geometries", Computer Physics Communications, 178, (12), 894-914, (2008).
- D. Groen, J. Hetherington, H. B. Carver, R. W. Nash, M. O. Bernabeu, P. V. Coveney, "Analyzing and Modeling the Performance of the HemeLB Lattice-Boltzmann Simulation Environment", Journal of Computational Science, (2013), 4 (5), 412–422.
- A. Patronis, R. A. Richardson, S. Schmieschek, B. J. Wylie, R. W. Nash and P. V. Coveney, "Modelling Patient-Specific Magnetic Drug Targeting within the Intracranial Vasculature", Frontiers of Physiology, 9:331 (2018).

Image References

- Slide 3 - http://www.sciencekids.co.nz/pictures/humanbody/humanorgans.html
- Slide 5 - https://stanfordhealthcare.org/medical-tests/p/pet-mri-scan.html
- Slide 7 - http://www.netlib.org/utk/people/JackDongarra/PAPERS/sunway-report-2016.pdf
- Slide 12 - A. Patronis, R. A. Richardson, S. Schmieschek, B. J. Wylie, R. W. Nash and P. V. Coveney, Frontiers of Physiology, 9:331 (2018).
- Slide 13/14 - S.S. Esfahani *et al.* 2019 - *In Preparation*
- Slide 18/23 - D. Groen, D. Abou Chacra, R. W. Nash, J. Jaros, M. O. Bernabeu, P. V. Coveney, EASC 2014, LNCS 8759, 28–38 (2015)
- Slide 21 - A.R.G Harwood and A.J. Revell, Advances in Engineering Software (2017), Vol. 104, pp. 38-50.

Q&A

To pose a question, you can write your question in the “Questions” tab

Thank you for participating!

…don't forget to fill in our feedback questionnaire…

Visit the CompBioMed website (www.compbiomed.eu/training)
for a full recording of this and other webinars,
to download the slides
and to keep updated on our upcoming trainings

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 675451

The series is run in collaboration with: VPH Institute Building the Virtual Physiological Human