ARCHER eCSE Final Report

eCSE Number:	eCSE01-010
eCSE Title:	Adding a resolved deformable particle model to a highly-parallel blood flow solver for sparse vascular networks
Date of Submission:	15 February 2016
PI and Co-Is:	Miguel O. Bernabeu, Timm Krüger, Peter V. Coveney, James Hetherington, Bruno Silva
Technical staff member(s):	Mayeul d'Avezac, Owain Kenway, Ian Kirker, Jens Nielsen, Gary Macindoe
Author(s) of this document:	Miguel O. Bernabeu, Timm Krüger
Project start date:	1 September 2014
Project completion date:	31 August 2015
Total number of funded project months:	12

1 Publishable Summary

1.1 Achievement of objectives

The main objective of the project was to efficiently implement a model of deformable particles into the HemeLB flow solver. This will allow us to model blood as a dense suspension of deformable particles and facilitate the theoretical study of advanced aspects of haemorheology, oxygen transport, and cell trafficking in realistic vascular networks.

The main components to be implemented were: a) the immersed boundary method for fluid-structure interaction, b) a finite element model of particle membrane elastic deformation, and c) a procedure for simulation initialisation. An important challenge to be overcome in the project was developing a generic implementation of the previous algorithms that can scale up well in terms of domain complexity and particle volume fraction without undermining the excellent parallel scalability of HemeLB.

The objectives of the project were largely achieved (see Section 3 for a detailed report on each work package). Novel scientific results were generated and presented at international conferences and invited talks by the PI and co-Is of the grant.

1.2 Project description

HemeLB (Mazzeo & Coveney 2008) is an open source, highly-parallel, lattice-Boltzmann (LB) blood flow simulator originally developed at University College London, since 2007. One of HemeLB's unique features is being designed to run efficiently in sparse vascular networks, unlike similar LB codes where the degree of sparsity adversely affects computational efficiency. This fact (in addition to the embarrassingly parallel nature of the LB algorithm) makes HemeLB very well suited for the simulation of blood flow in large vascular trees using state-of-the-art supercomputing resources.

HemeLB has been extensively used for the simulation of blood flow in the neonatal mouse retina (a common experimental model for the study of angiogenesis, *i.e.* the process of vessel formation from pre-existing vasculature). This has been facilitated by recent advances in imaging technology that have allowed us to image and reconstruct retinal vascular networks at a level of detail never seen before (see Figure 1). In (Bernabeu et al. 2014; Franco et al. 2015; Franco et al. 2016), we presented *in silico* measurements of wall shear stress (WSS) in the retinal vasculature and provided evidence of the previously hypothesised relationship between haemodynamic forces and vascular remodelling during angiogenesis.

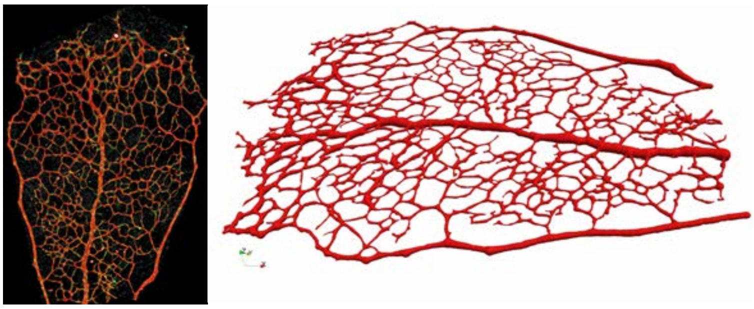
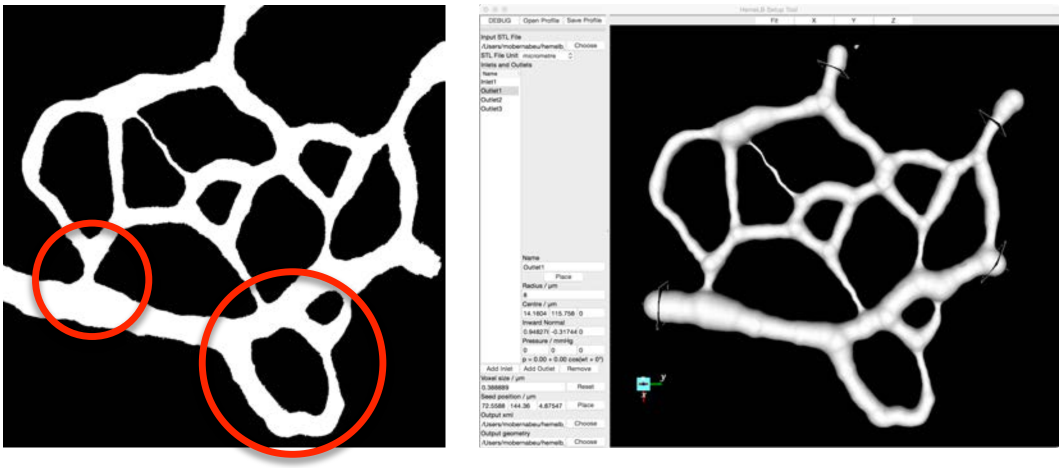


Figure 1: (left panel) retinal plexus samples collected from 6-day-old mouse pups and stained with endothelial luminal marker, (right panel) reconstructed three-dimensional luminal surface ready to be used for HemeLB simulation, from (Bernabeu et al. 2014).

In those works, HemeLB modelled blood as a shear-thinning homogeneous fluid rather than a complex particle suspension. This simplification is suitable for the study of haemodynamics in larger vessels (*i.e.* those with diameters larger than a few hundred micrometres (Lei et al. 2013)). However it fails to capture important haemorheological features when applied to the simulation of blood flow in capillaries, especially as vessel calibre gets closer to, or even smaller than, the typical red blood cell (RBC) diameter of ~8 micrometres. This shortcoming is preventing us from achieving a higher degree of fidelity in our WSS estimates in the previous applications. On the one hand, (Xiong & Zhang 2010) measured up to a 20% increase in WSS due to the presence of RBCs in a simplified model of microcirculation when compared to the homogeneous approach. On the other hand, the well-characterised Fåhræus-Lindqvist effect predicts a sharp decrease in apparent viscosity with decreasing diameter in microcirculation. Being able to simulate both effects in a mechanistic way would provide a much more accurate picture of the WSS experienced by vessels. Furthermore, it would also allow us to derive reliable constitutive models that can be used in the homogeneous case, at the expense of some accuracy, when computational requirements do not allow explicit RBC simulation.

In the current project, we extended HemeLB with a model of deformable particles. From a numerical point of view, the project involves the implementation of: a) the immersed boundary method for fluid-structure interaction, b) a finite element model of particle membrane elastic deformation, c) a procedure for simulation initialisation based on a given target RBC volume fraction (haematocrit). Algorithms for these tasks have been proposed in the literature (see (Krüger 2012) and references therein) and have been successfully applied to the simulation of deformable particle suspensions flowing in idealised geometries (Krüger et al. 2013; Kaoui et al. 2012; Krüger et al. 2011; Kaoui et al. 2013). However, to the best of our knowledge, this project represented the first attempt to achieve similar results in sparse domains like the one presented in Figure 1. This represents a substantial leap forward for the simulation of blood flow in microvasculature and enables for the first time the theoretical study of advanced aspects of haemorheology, oxygen transport, and cell trafficking in realistic vascular networks.

Our preliminary simulations to date have focused on the study of the collective dynamics of dense RBC suspensions in complex vessel geometries. Briefly, we have analysed how the choice of rheology model (shear-thinning homogeneous vs particle suspension) affects the predicted wall shear stress (WSS) levels in our models of flow in the developing vasculature (Figure 2).



Input STL File
/Users/mobernabeu/hemelb
STL File Unit micrometre
Inlets and Outlets
Name
Inlet1
Outlet1
Outlet2
Outlet3
Name
Outlet1
Radius / μm
8
Centre / μm
14.1604 115.758 0
Inward Normal
0.948271 -0.31744 0
Pressure / mmHg
0 0 0
p = 0.00 + 0.00 cos(ωt + 0°)
Voxel size / μm
0.388889
Seed position / μm
72.5588 144.36 4.87547
Output xml
/Users/mobernabeu/hemelb
Output geometry
/Users/mobernabeu/hemelb

Figure 2. Left panel: binary image representing the luminal space (white) against background tissue (black) of a subset of a 6-day-old mouse retinal vasculature. Red circles indicate regions of interest. Right panel: HemeLB's setup tool where the flow simulation is defined.

Our preliminary simulations show how the WSS predicted by the shear-thinning homogeneous model greatly deviates from the its particle suspension counterpart in the presence of vessel branches with sharp angles (Figure 3) and in situations of RBC depletion occurring due to an increase in vessel segment resistance (Figure 4). These preliminary results were accepted for oral presentation at the VI International Conference on Computational Bioengineering in Barcelona, Spain (ICCB 2015).

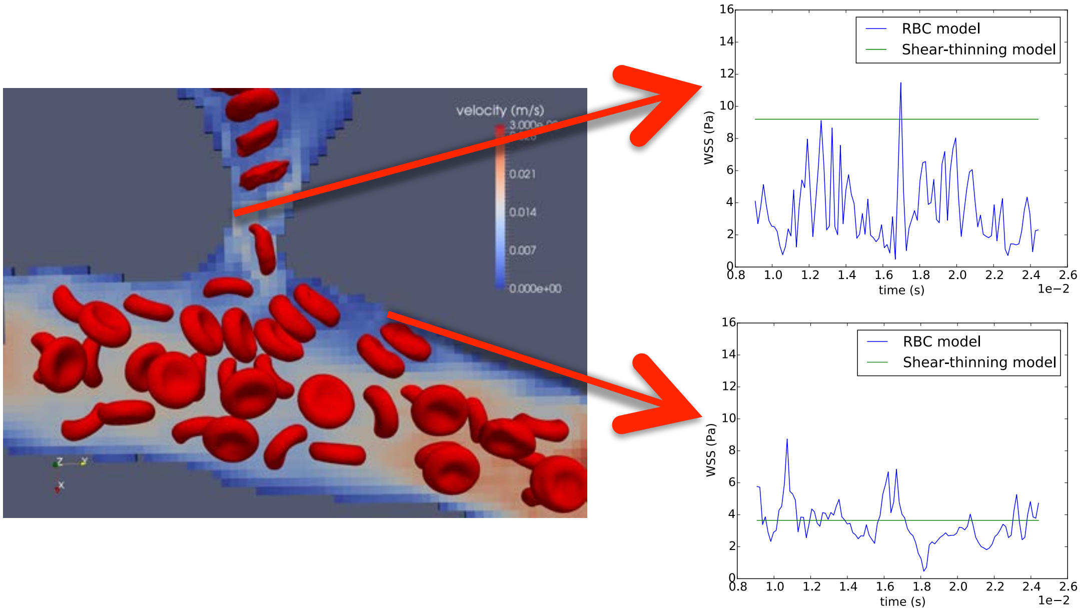


Figure 3. Left panel: Snapshot of the simulation at one of the regions of interest. RBC membranes are rendered in red. The flow field is plotted at the plane perpendicular to the viewer and containing the vessel centreline. Right panels: WSS traces at two points of interest (indicated by the origin of the red arrows). In the mother vessel (bottom) both RBC and homogeneous model have a good agreement. However, in the vicinity of a sharp turn (top), the homogeneous model greatly overestimates WSS due to its inability to capture complex rheological effects. In both cases, the homogeneous model fails to capture the temporal complexity of the WSS signal recovered with the RBC model.

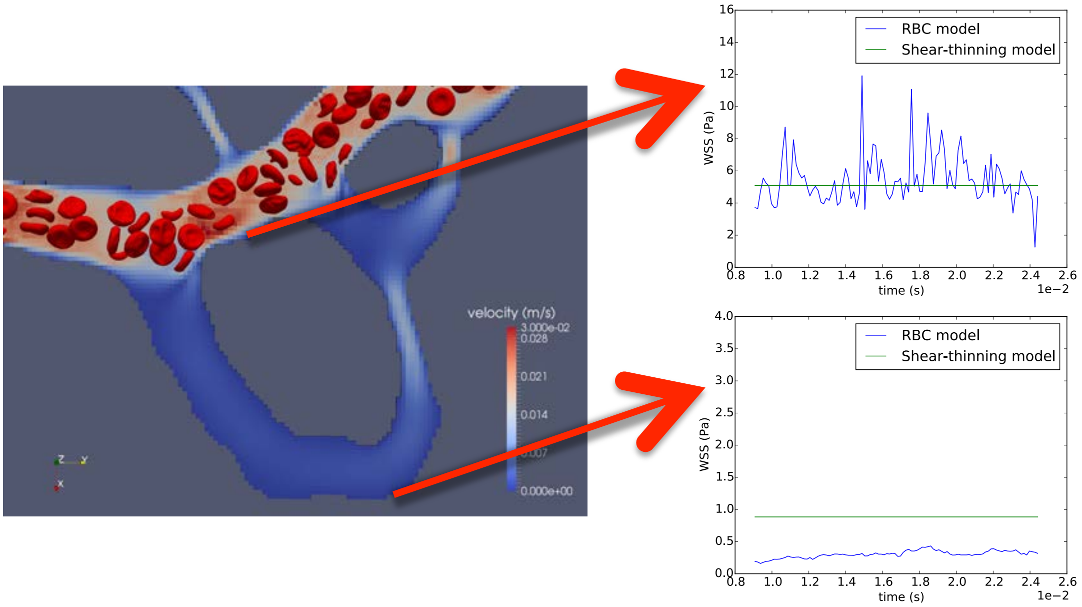


Figure 4. Left panel: Snapshot of the simulation at one of the regions of interest. RBC membranes are rendered in red. The flow field is plotted at the plane perpendicular to the viewer and containing the vessel centreline. Right panels: WSS traces at two points of interest (indicated by the origin of the red arrows). We see again that in the mother vessel (top) both RBC and homogeneous model have a good agreement. However, in the presence of a loop (bottom), the RBC model predicts a much lower WSS level due to the RBC depletion effect associated with the increased flow resistance of the loop.

1.3 Summary of the software

The main code to be worked on during the project is HemeLB (Mazzeo & Coveney 2008). HemeLB is an open source, highly-parallel, lattice-Boltzmann (LB) blood flow simulator originally developed at the Centre for Computational Science (CCS), University College London (UCL), since 2007. HemeLB's main codebase currently receives contributions from additional UK research groups at The University of Edinburgh and Brunel University, London. HemeLB has supported various publications by the project PI and co-I (Bernabeu et al. 2013; Nash et al. 2014; Bernabeu et al. 2014; Franco et al. 2015) as well as external collaborators (Chen et al. 2012; Groen et al. 2013). Furthermore, HemeLB was one of the six exemplar projects in the EU-FP7 CRESTA project concerning the development of exascale systemware, tools, and applications.

HemeLB is a C++/Python application. The main computational kernel is written in C++ and all the pre- and post-processing tools are written in Python. The code is version controlled with Git with a release version being pushed to GitHub (https://github.com/UCL/hemelb). HemeLB is developed using a test-driven approach. Our continuous integration system runs unit and regression tests (currently 136 tests making 448 assertions) after each commit and also every night across a range of supported platforms. Software development is managed using a Scrum process with fortnightly sprints. The HemeLB development team is made of a core of 3-4 developers at UCL and The University of Edinburgh, and a number of external collaborators. HemeLB has successfully run in the following top 500 machines: ARCHER, HECToR, SuperMUC, and Kraken. Figure 2 presents a parallel scalability analysis performed in the HECToR Cray XE6 machine and published in (Groen et al. 2012).

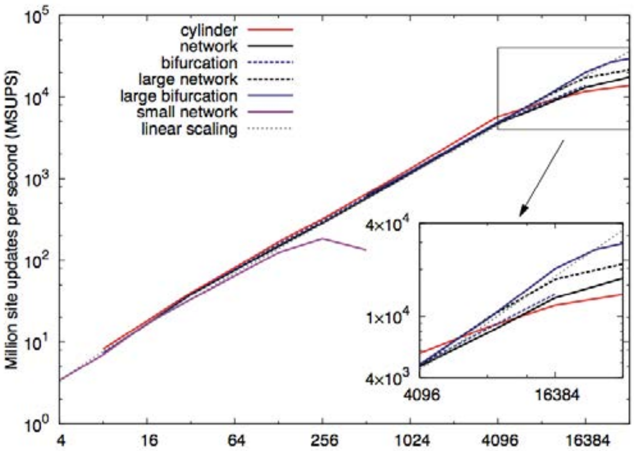


Figure 2: From (Groen et al. 2012) *Lattice site updates per second (LSUPS) as a function of the number of cores used for simulations run on the HECToR Cray XE6 machine. Simulations run with six different fluid domains, ranging from 77k fluid sites (small network) to 44M fluid sites (large network). For a big enough problem, HemeLB scales linearly up to 16k cores. Unpublished results demonstrate further linear scaling up to 64k cores for big enough problems.*

The following HemeLB development infrastructure improvements were undertaken in the context of the project:

1) HemeLB development was ported to Github. This facilitates multisite development (London and Edinburgh) and code review (through pull request feature).
2) HemeLB's continuous integration was ported to the UCL Jenkins service: we now test with multiple compilers and OS (OS X, desktop linux and cluster linux). We added the ability to test Github branches.
3) A Slack channel was created for developer communication and code design discussions.
4) Support for parallel unit testing was added. To date all unit tests were run sequentially, therefore we could not test against communication bugs, deadlocks, etc. A user would only discover them once the code is run in production in a cluster, which makes debugging difficult and wastes CPU hours.
5) We enabled the use of C+11 features. Some of its features had been manually added to HemeLB and had to be maintained by the team, now it uses the equivalent standard implementation (STATIC_ASSERT and boost:shared_ptr() -> std::shared_ptr()).