International Journal of Computer Assisted Radiology and Surgery manuscript No.
(will be inserted by the editor)

Lattice-Boltzmann Interactive Blood Flow Simulation Pipeline

Sahar S. Esfahani [1] · Xiaojun Zhai [2] · Minsi Chen [3] · Abbes Amira [4] · Faycal Bensaali [1] · Julien AbiNahed [5] · Sarada Dakua [5] · Georges Younes [5] · Abdulla Baobeid [5] · Robin A. Richardson [6] · Peter V. Coveney [6]

Received: date / Accepted: date

Abstract
Purpose Cerebral aneurysms are one of the prevalent cerebrovascular disorders in adults worldwide and caused by a weakness in the brain artery. The most impressive treatment for a brain aneurysm is interventional radiology treatment, which is extremely dependent on the skill level of the radiologist. Hence, accurate detection and effective therapy for cerebral aneurysms still remain important clinical challenges. In this work, we have introduced a pipeline for cerebral blood flow simulation and real-time visualization incorporating all aspects from medical image acquisition to real-time visualization and steering.

Methods We have developed and employed an improved version of HemeLB as the main computational core of the pipeline. HemeLB is a massive parallel lattice-Boltzmann fluid solver optimized for sparse and complex geometries. The visualization component of this pipeline is based on the ray marching method implemented on CUDA capable GPU cores.

Results The proposed visualization engine is evaluated comprehensively and the reported results demonstrate that it achieves significantly higher scalability and sites updates per second, indicating higher update rate of geometry sites' values, in comparison to the original HemeLB. This proposed engine is more than two times

✉ Abbes Amira
abbes.amira@dmu.ac.uk

[1] College of Engineering, Qatar university, Doha, Qatar

[2] School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK

[3] Department of Computer Science, University of Huddersfield, Huddersfield, UK

[4] Faculty of Computing, Engineering and Media, De Montfort University, Leicester, UK

[5] Department of Surgery, Hamad Medical Corporation, Doha, Qatar

[6] Centre for Computational Science, University College London, London, UK

faster and capable of 3D visualization of the results by processing more than 30 frames per second.
Conclusion A reliable modelling and visualizing environment for measuring and displaying blood flow patterns in vivo, which can provide insight into the hemodynamic characteristics of cerebral aneurysms, is presented in this work. This pipeline increases the speed of visualization and maximizes the performance of the processing units to do the tasks by breaking them into smaller tasks and working with GPU to render the images. Hence, the proposed pipeline can be applied as part of clinical routines to provide the clinicians with the real-time cerebral blood flow related information.

Keywords Cerebral Aneurysm · Pipeline · Lattice-Boltzmann · GPU · Visualization

1 Introduction

A cerebral aneurysm (also known as intracranial aneurysms or brain aneurysms) is a malformation in a brain vessel caused by a weakness in the vessel wall as shown in Fig. 1. Accurate diagnosis, as well as efficient treatments of a brain aneurysm in its early stages, are crucial for the patients who suffer from this brain disorder. However, using intra-aneurysmal coils along with the stents in interventional radiology treatments is the most effective and common approach deployed to treat cerebral aneurysms [1]; it is still a clinically challenging intervention with high failure rates. This is due to the dependence of this approach on the skill level of the radiologist in recognizing the vascular geometry and estimating blood flow related information from the 2D and/or 3D medical images. Hence, providing clinicians with vascular geometries and fluid flow mechanical parameters such as velocity and pressure retrieved from the blood flow simulation can help them to have the best prognosis and treatment for these patients.

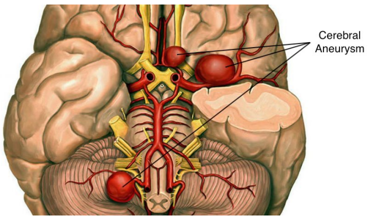


Fig. 1: Brain aneurysm [2].

Recent developments in Computational Fluid Dynamic (CFD) techniques have led them to be widely used in fluid flow simulations. Lattice-Boltzmann Method (LBM) is a class of CFD's designed for time-dependent simulation of large scale systems by using a parallel implementation [3]. LBM can be efficiently implemented on various parallel architectures from Field Programmable Gate Arrays (FPGAs) [4] to General Purpose Graphical Processing Units (GPGPUs) [5] and supercomputers [6].

HemeLB is a fluid solver which utilizes parallel LBM to compute and analyze the bio-mechanical modelling of the large scale fluid flow in complex geometries [3]. The original version of HemeLB uses CPU cores for the LBM computations and rendering task and has been successfully implemented on open academic High-Performance Computing (HPC) platforms such as ARCHER [7], Blue Waters [8], HECTor, and SuperMUC [9]. The main objective of designing HemeLB is to enable clinicians and radiologists to examine the cerebral blood flow behavior in the human body [7, 10].

Recently, some research has been conducted to optimize and deploy HemeLB on cost-efficient platforms such as FPGAs and GPUs to speed up the simulation for potentially applying the framework in medical environments [11]. Moreover, the visualization solution for HemeLB introduced by Mazzeo et al. [12] can help radiologists and surgeons explore the results in real-time. This visualization pipeline simulates and visualizes the cerebral blood flow interactively. In their solution, the rendering is implemented using an optimized ray casting technique integrated into HemeLB. By employing this solution, the flow field data are rendered in situ on the processors responsible for running the LBM and then are sent to a host computer equipped with a graphical frontend. In fact, flow field simulation and rendering are performed simultaneously on CPU cores and then rendered frames are output to be visualized either locally or remotely.

Despite Mazzeo et al. [12] having integrated a real-time visualization engine into the original version of HemeLB, they do not provide a complete solution for applying HemeLB as part of clinical routines. It is essential to develop a powerful pipeline for real-time monitoring of the hemodynamic variables of the cerebral aneurysms; a pipeline which can be fully under the control of clinicians and generate the real-time visual results from the input medical images. Additionally, the performance of their visualization engine is limited to the CPU cores responsible for the rendering task. With rapid improvement of the resolution of medical images over recent years, a powerful visualization engine is required to serve clinicians with 3D description of the data. Considering the fact that analyzing and processing the medical images with high dimension are computationally expensive, GPU's are more suitable for rendering and processing medical images [13, 14].

In this paper, we introduce an efficient integrated patient-specific cerebral aneurysm blood flow simulation and real-time visualization framework. The proposed pipeline covers all the stages required for blood flow simulation from image acquisition to result visualization in the clinical time frame. We used HemeLB fluid solver as the main computational core of the pipeline to investigate the cerebral blood flow behavior. In this study, we focus on visualizing pressure, velocity, and wall shear stress which are important parameters in brain aneurysm diagnosis. A real-time GPU rendering engine based on the original HemeLB code has also been introduced. This visualization component is implemented on Compute Unified Device Architecture (CUDA) capable GPU's with the aim of accelerating the image rendering. In addition, a set of comprehensive tests using data from real patients and phantom models has been carried out to evaluate the accuracy and performance of the presented rendering engine. The reported results demonstrate that the proposed GPU rendering achieves better performance in comparison to the original HemeLB rendering and hence our pipeline is assuredly appropriate for the clinicians to conveniently discover and analyze the 3D simulation outputs from the input medical images in real-time.

This paper begins with introducing the various components of the pipeline in Section 2, then goes on to focus on the visualization approach in Section 3. The pipeline implementation and performance analysis are presented in Section 4 and 5, respectively. Finally, the paper is concluded in Section 6.

2 Blood Flow Simulation and Visualization Pipeline

This integrated pipeline provides users with real-time visualized results of the hemodynamics in brain aneurysms generated from the input Digital Imaging and COmmunications in Medicine (DICOM) images. Every stage in the process of cerebral aneurysm treatment, from image acquisition and segmentation to result visualization and steering, is addressed in this environment. Fig. 2 illustrates the pipeline with different stages. They include medical image acquisition, image segmentation and meshing, simulation configuration, computational fluid dynamics simulation, and results visualization. A visualization platform design with the aim of providing clinicians with a smooth and interactive environment is also included. This platform helps users access each component of the pipeline conveniently. The pipeline components are described in the following sections.

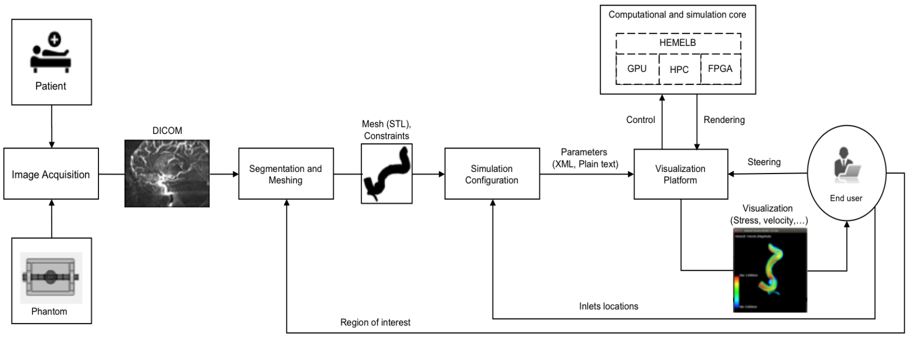


Fig. 2: Proposed real-time blood flow simulation and visualization pipeline.

2.1 Medical Image Acquisition

The first and early step in cerebral aneurysm detection and therapy planning is providing medical images of the brain under investigation. The major imaging methods used for neuro-imaging of cerebral aneurysms are: Computed Tomography Angiography (CTA), Magnetic Resonance Angiography (MRA), Digital Subtraction Angiography (DSA), and 3-Dimensional Rotational Angiography (3DRA).

CTA: If the clinical suspicion level is high, but there is negative non-contrast CT, the CTA is useful to detect the ruptured brain aneurysms [15]. However, current multi-detector scanners can reliably detect aneurysms greater than 4 mm with 100% sensitivity [16, 17].

MRA: This is more advantageous than CTA since it does not have ionization and is able to obtain images even without intravenous contrast agents. However, Contrast Enhanced MRA (CE-MRA) is found to be superior to Time Of Flight TOF MRA (TOF MRA does not use contrast agents) because it can eliminate flow-related artifacts [18]. 3T MRA can better visualize small branch vessels compared to CTA. Presently, machines with 1.5 T and 3 T have spatial resolution 1 mm [19] and 0.6-0.7 mm [20], respectively. However, because of higher cost and longer acquisition times, MRA is not preferred.

DSA: Majority of the ruptured aneurysms are found small [21] and DSA is useful to detect such aneurysms when CTA and MRA miss them. DSA too requires ionizing radiation and iodinated contrast like CTA. However, the procedure is invasive requiring highly skilled operator in order to guide the catheter to the major intracranial vessels.

3DRA: This has higher sensitivity and can detect aneurysms who sizes are less than 3 mm and can go until 0.5 mm [22]. Van Rooji et al. [22] have reported that 94 additional aneurysms were detected that were missed by CTA and MRA. 27 out of these 94 aneurysms were even missed by DSA.

Therefore, 3DRA is presently the preferred choice by the clinicians as compared to DSA, MRA, and CTA. Accordingly, we have used 3DRA images as the input in the current pipeline.

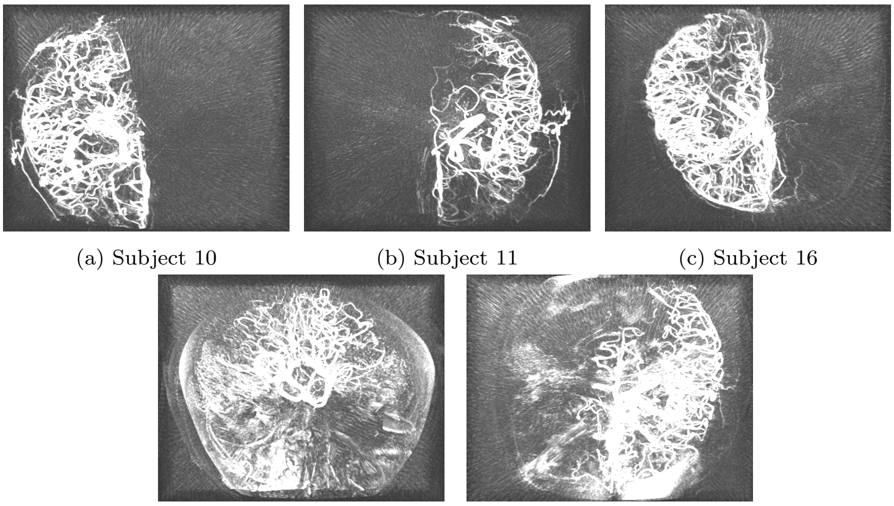
The datasets used in this paper have been obtained from two different imaging systems: Artis Axiom Angiography Interventional system by Siemens, and the Allura FD20/15 Interventional Neuroradiology X-ray system by Philips, located at Hamad Medical Corporation (HMC) in Qatar. On average, there are 400 slices in each dataset acquired along the long axes of the subjects [23]. The average slice thickness, matrix size, and pixel spacing is 0.29 mm, 512 × 512, and 0.29 mm × 0.29 mm, respectively. Five subjects of 3DRA carefully selected by HMC clinicians are shown in Fig. 3.

These input images can be taken from real patients or phantom model of a 3D printed generic aneurysm which is used in this work to evaluate the results. Fig. 4 illustrates the schematic of the phantom setup. In this setup, fluid flows through the phantom and circulates back to the reservoir tank through plastic tubes. A medical Y connector allows the secure access of the microcatheter and the pressure sensor.

2.2 Image Segmentation and Meshing

Two different segmentation methods have been applied at this stage. The first method is based on the technique presented by Zhai et al. [24]. In their study, aneurysm segmentation is carried out using wavelet transform and hard thresholding algorithm. This segmentation technique consists of the following six steps and is represented as Algorithm 1 and 2 in Online Resource 1.

1. Normalize intensity propagation of 3DRA image to [0-255].
2. Apply Haar wavelet decomposition on the normalized slice image.
3. Apply hard thresholding on the approximated coefficients.


(a) Subject 10 (b) Subject 11 (c) Subject 16

(d) Subject 19 (e) Subject 23

Fig. 3: 3DRA images of five patients with brain aneurysm.

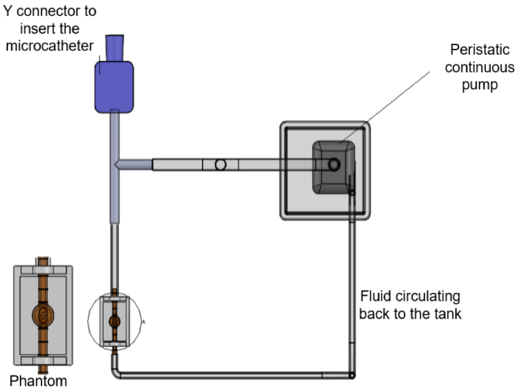

Y connector to insert the microcatheter
Peristatic continuous pump
Fluid circulating back to the tank
Phantom

Fig. 4: Schematic of the phantom setup.

4. Reconstruct each slice by wavelet detail and thresholded approximate coefficients.
5. Convert the reconstructed slice to binary segmented mask by a threshold.
6. Concatenate and process the segmented masks of slices to build the volume of vessels.

The second technique is an efficient method based on simple thresholding and post-processing the results [23]. Automated vessels segmentation, semi-automated aneurysm detection, and automatic boundary and inlet detection from the input 3DRA image, are all addressed in this method.

The output of this pipeline stage is a STereoLithograph (STL) file that represents the aneurysm mesh clipped at the inlet and outlets. The mesh files generated from subject 10 (see Fig. 3) by using two aforementioned segmentation techniques are illustrated in Fig. 5.

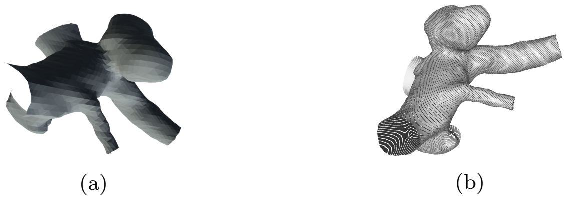


Fig. 5: STL meshes generated at the image segmentation component of the pipeline; a) rendered mesh using segmentation method described by Zhai et al. [24]; b) mesh file rendered by applying slicing planes [23].

2.3 Simulation Configuration

Once the STL mesh is obtained, it is used to generate the configuration file for simulation. Later, this file is applied in the hemodynamic simulation stage carried out using HemeLB. HemeLB is a Message Passing Interface (MPI) based lattice-Boltzmann solver optimized for highly sparse geometries found in the cerebral vasculature [3]. It can efficiently model flows in cerebral arteries using up to 32,768 cores [9]. HemeLB utilizes a weighted domain decomposition approach to reduce the overheads of various boundary conditions, where multiple levels of boundary conditions have been generated to minimize the computational costs, e.g. compute-intensive boundary and in/outflow conditions. In addition, HemeLB is flexible enough to allow the user to achieve key flow properties, for example, pressure, velocity, and wall shear stress at predefined time intervals. All those parameters are integrated into a property-extraction framework and generated together with the geometry simulation files [25].

2.4 Computational and Simulation Core

We applied HemeLB as the pipeline's main computational and simulation core. The flow rates, pressures and shear stresses throughout the vascular structure are computed by the LBM implemented in HemeLB. LBM is a fast simulation technique for large and complex fluid systems [3]. In this method, it is considered that the propagation and collision processes of local particle distributions are performed over a discrete lattice mesh. The streaming and collision equations are described in detail by Zhai et al. [4]. The parallelized LBM is deployed in HemeLB as the fluid solver in which fluids are represented with particle distribution functions. We applied a three-dimensional lattice with 15 discrete velocity directions (D3Q15) in our pipeline as shown in Fig. 6.

Various Boundary Conditions (BC) can be considered to implement HemeLB. Two examples of these BCs are Bouzidi-Firdaouss-Lallemand (BFL) for the interpolated wall collision BC presented by Bouzidi et al. [26] and Ladd iolets for velocity inlet BC [27]. Applying a topology-aware two-level domain decomposition in HemeLB ensures a good workload distribution for parallel implementation. Furthermore, reduction of redundant operations, pattern regularity improvements and enhancement in intra-machine communication are the other improvements made in HemeLB.

2.5 Visualization Platform

As mentioned in Section 1, this pipeline is developed to support clinical decisions in brain aneurysm detection and therapy. Therefore, the users have to be able to launch the pipeline in a smooth and interactive environment. The visualization platform allows clinicians to control the simulation operations and monitor the 3D results in real-time. Interventional radiologists can use this platform to upload 3DRA images taken from the patient, then move between the different components of the pipeline and explore the 3D results. Further details of real-time 3D visualization are provided in the next section.

3 Real-time Visualization

A key factor in developing an efficient clinical simulation environment is real-time visualization of the results instead of generating and storing frames on disk and post-processing them. Volume rendering plays a crucial role in immediate and accurate visualization of the simulation results in various scientific areas [28]. In this pipeline, the real-time visualization is implemented in situ by GPU cores with CUDA capabilities. The CUDA cores execute the direct volume rendering kernel based on the ray marching method [29] which is represented in Algorithm 1.

Fig. 7 illustrates the architecture of the multi-core simulation and real-time visualization in our pipeline. As depicted in this diagram, the compute nodes are responsible for the LBM computations. Each LB node computes and caches the LB properties of all lattice sites dedicated to it and then transfers the LB data to the master node. Even though this node does not participate in LBM computations, it manages the view steering, schedules the lattice data transfer and

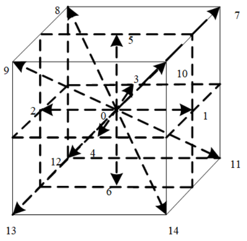


Fig. 6: Lattice node of D3Q15 model [11].

Algorithm 1 The CUDA ray casting kernel function pseudo code.

```
Input: VolumeData, TransferFunction, ImageWidth, ImageHeight
Output: OutputColour
procedure VolumeRenderKernel($x, y, S, \delta s, \rho, \mathbf{M}$)    ▷ $x, y$: pixel position; $S$: ray marching step; $\delta s$: ray step size; $\rho$: voxel density; $\mathbf{M}$: view matrix
    $OutputColour \leftarrow 0$
    ▷ $(u, v)$: normalized coordinates of the screen
    $u \leftarrow x/ImageWidth * 2 - 1$
    $v \leftarrow y/ImageHeight * 2 - 1$
    ▷ Calculate ray in world(view) space
    $\mathbf{o} \leftarrow (0, 0, 0)$
    $\mathbf{d} \leftarrow (u, v, -2)$
    $\mathbf{d}' \leftarrow \mathbf{M}^{-1}\mathbf{d}$
    $Ray \leftarrow (\mathbf{o}, \mathbf{d}')$
    $t \leftarrow$ IntersectVolume($VolumeData, Ray$)    ▷ Find intersection with box
    ▷ March along ray from front to back
    if $0 \leq t < MAX$ then
        $\mathbf{p} \leftarrow \mathbf{o} + \mathbf{d}'t$    ▷ Nearest voxel position; Ray intersects with the nearest voxel in Volume
        $\delta\mathbf{d} \leftarrow \mathbf{d}'\delta s$    ▷ Step vector for ray cast
        $s \leftarrow 0$
        while $s < S$ do
            $Value \leftarrow$ SampleVolume($\mathbf{p}$)
            $Colour \leftarrow$ TransferFunction($Value$)    ▷ Transfer $Value$ to RGBA
            $Colour \leftarrow Colour \cdot \rho$
            $OutputColour \leftarrow$ AlphaBlend($OutputColour, Colour$)    ▷ Integrate voxel colors
            $\mathbf{p} \leftarrow \mathbf{p} + \delta\mathbf{d}$    ▷ March along the ray by $\delta\mathbf{d}$
            $s \leftarrow s + 1$    ▷ Next step
        end while
    end if
end procedure
```

also shares the lattice data with CUDA capable GPU cores to perform rendering simultaneously. The communication between the visualization client and compute nodes is performed by the MPI exploited in HemeLB. As it is represented in Fig. 7, the internal architecture of the visualization client is a three-tier system. The front end layer is an OpenGL application and the middle tier is the host which stores steering parameters and cached lattice data. Finally, the top tier is the rendering layer where all the voxels are stored and CUDA cores execute volume rendering kernels.

In real-time visualization, the number of memory transfers is critical. In our pipeline, this includes transmission of LB properties and steering parameters. We applied a two-level memory access optimization in order to decrease the number of memory transfers and enhance the throughput between the GPU and CPU [30].

Important lattice properties such as pressure, velocity, and shear stress are visualized in our visualization client. The 3D display of the results can be illustrated in two different modes (see Fig. 8): 1) multi-view window; and 2) full separate window for each lattice property. In Fig. 8(b), the dynamic color map has been set based on the minimum and maximum values of each simulated lattice property.

In this pipeline, visualization steering has also been integrated to allow users to adjust the rendering parameters interactively and hence to reduce generation time of the results. The steerable parameters for the visualization include model rotation, zooming, and adjusting the scale and offset of the transfer functions. A

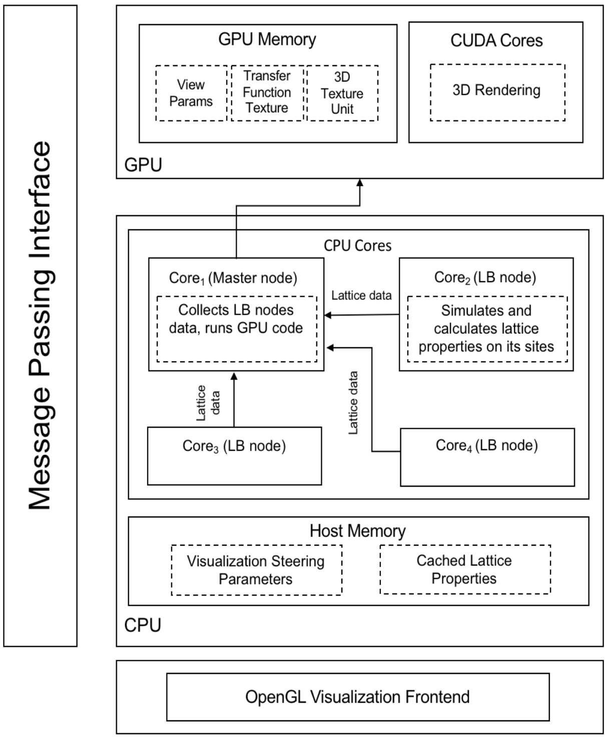

Message Passing Interface
GPU Memory
View Params
Transfer Function Texture
3D Texture Unit
CUDA Cores
3D Rendering
GPU
CPU Cores
Core$_1$ (Master node)
Collects LB nodes data, runs GPU code
Lattice data
Core$_2$ (LB node)
Simulates and calculates lattice properties on its sites
Lattice data
Lattice data
Core$_3$ (LB node)
Core$_4$ (LB node)
Host Memory
Visualization Steering Parameters
Cached Lattice Properties
CPU
OpenGL Visualization Frontend

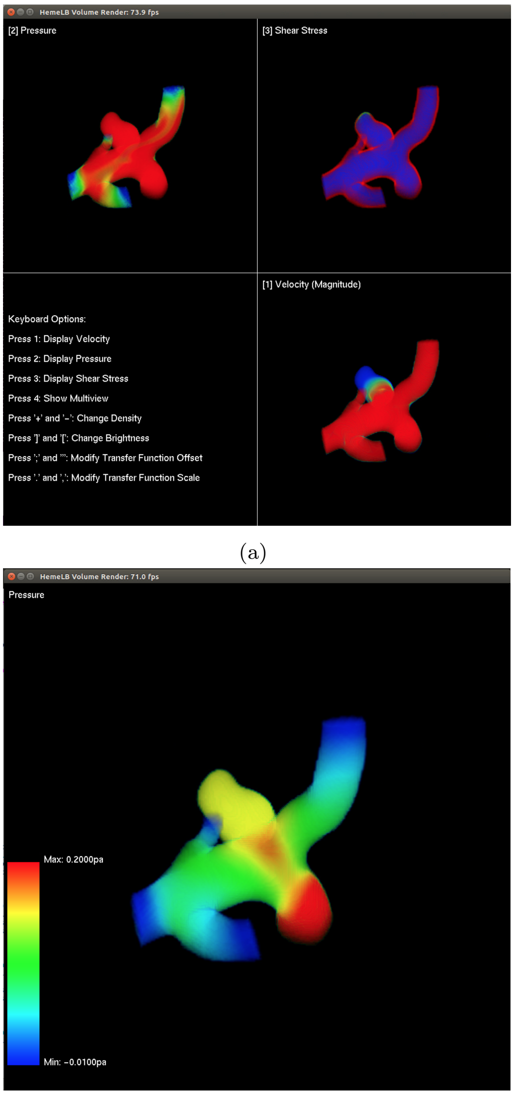
Fig. 7: The three-tier architecture of the proposed visualization client; running simulation with 4 cores.

set of keyboard options listed in Fig. 8(a) enables users to modify various rendering parameters and also choose the appropriate display mode.

4 Implementation

4.1 Hardware Platform

The pipeline has been benchmarked on an Exxact Tensor TWS-289059-DPN workstation which is known as a deep learning NVIDIA GPU solution [31]. Table 1 represents the overview of this workstation.



Keyboard Options:
Press 1: Display Velocity
Press 2: Display Pressure
Press 3: Display Shear Stress
Press 4: Show Multiview
Press ’+’ and ’–’: Change Density
Press ’]’ and ’[’: Change Brightness
Press ’;’ and ’’’: Modify Transfer Function Offset
Press ’.’ and ’,’: Modify Transfer Function Scale

(a)

(b)

Fig. 8: Visualization window; a) Multi-view window splited into 4 sub windows: one for the visualization instruction as shown in the bottom-left and three for the velocity, pressure, and shear stress displayed in bottom-right, top-left and top-right sub windows, respectively; (b) Single-view window displaying pressure values with relevant scale.

4.2 Phantom Setup

Fig. 9 shows the phantom setup in the catheterization laboratory at HMC. The phantom is designed as a generic aneurysm with 4.0 mm artery diameter and

Table 1: Overview of the TWS-289059-DPN workstation.

CPU	1× Intel Xeon(R) CPU E5-2623 v4 @ 2.60GHz with 8 cores
RAM	128GB DDR4 SDRAM
GPU	4× NVIDIA GeForce GTX 1080 Ti 11 GB GDDR5X PCIe 3.0 x16 packs 3584 CUDA cores
Storage	1.0TB SSD and 4.0TB HDD
OS	Ubuntu 16.04 LTS 64-bit

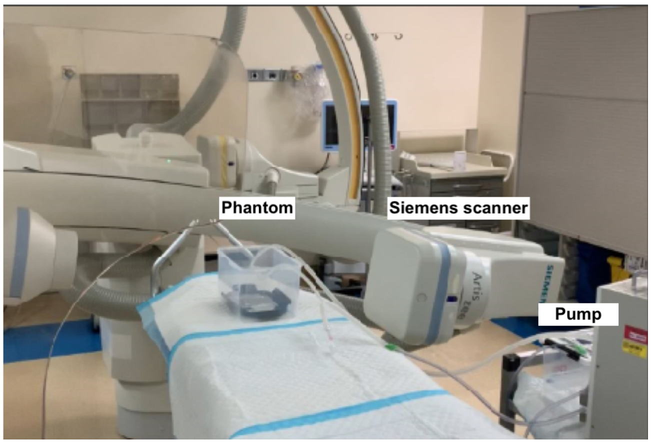


Fig. 9: In vitro setup at the catheterization laboratory at HMC.

aneurysmal sac diameter of 18.0 mm. This phantom is 3D printed using a laser-curing StereoLithography Apparatus SLA Moai Printer (Peopoly, USA [32]) with a layer height of 50 microns. For secure placement, the 3D printed phantom is fixed on acrylic sheets and placed under the imaging C-arm. It is connected with an EcoDrive continuous peristaltic pump (Olympus, Japan [33]) which maintains a constant flow rate of 1.8 liters/minute. To prevent interference with the imaging device, the pump is placed in a distance from the phantom which is constructed from metallic components. The circulating fluid has a density of 1.05 gm/cc and compromises of a mixture of distilled water (84.5%), glycerin (7%), and Iodine-based Ioxehol contrast (8.5%).

Fluid pressure measurements are collected using the Volcano Functional Measurements System (VFMS) and the Verrata sensor (Philips, United States [34]). The pressure sensor is inserted within a microcatheter to collect the pressure measurements before and after stent deployment. A Pipeline Embolization Device (PED) (Medtronic, USA [35]) is used as a flow diverter stent. This metallic mesh stent is deployed at the neck of the aneurysm and has a nominal diameter of 4.00 mm and a length of 20 mm.

5 Performance Analysis

In this section, we have mainly analyzed the rendering performance of two different versions of HemeLB implementation: 1) Original HemeLB; 2) HemeLB-GPU version. The performance of the visualization client has been benchmarked using the five geometries from Fig. 3. Since brain aneurysm datasets are not publicly available, the clinicians at HMC hospital have carefully chosen these subjects for this study; the selected subjects represent the cases of all types suitable for the study, so the initial visualization results could be used for a better case study of diagnose purpose. Depending on the memory size of the target platform, the proposed framework can be prototyped for larger cases as well.

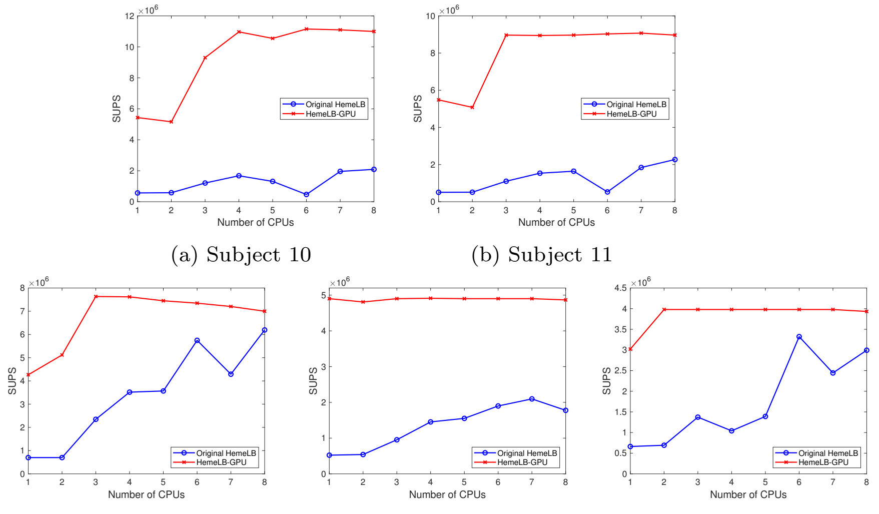
As stated before, the main motivation of this work is to increase rendering speed and design an algorithm to distribute the simulation and visualization workloads to different processing units. In original HemeLB implementation, the images are rendered using only CPU cores, where each CPU renders an image for a portion of the entire geometry, then all the images are sent to the master node to be recombined into a single image later. Similarly, in HemeLB-GPU version all the simulation data are sent back to the master node, but then the data are rendered by GPU cores. The overall performance comparison is shown in Fig. 10, where the number of rendered images and the total simulation steps are set to 25,000. In order to evaluate the performance of the proposed work and the update rate of sites' values, Sites Updates Per Second (SUPS) is measured based on

$$SUPS = \frac{N_{steps} \times N_{sites}}{T_s} \tag{1}$$

Where N_{steps} and T_s are the number of simulation steps and total time respectively, and N_{sites} indicates the number of geometry sites.

As it can be seen from Fig. 10, the results obtained from HemeLB-GPU version are significantly better than the original HemeLB. In addition, it is worth noting that the SUPS of the HemeLB-GPU version is improved considerably when increasing the cores from 1 to 3 for all the subject samples, however, it is more or less stable when the number of used cores is more than 4. In contrast to these results, the original HemeLB generates higher SUPS when more cores are involved in the computation for the subjects with more blocks (e.g. Subject 23 and Subject 16), but for the subjects with fewer blocks, SUPS increase gradually. Moreover, the HemeLB-GPU version renders 3 images (i.e. pressure, velocity, and wall shear stress) per request, however, the original HemeLB generates only one image (i.e. pressure) per request. On average, the HemeLB-GPU version takes about 11 ms for rendering an image, while the original HemeLB takes about 25 ms.

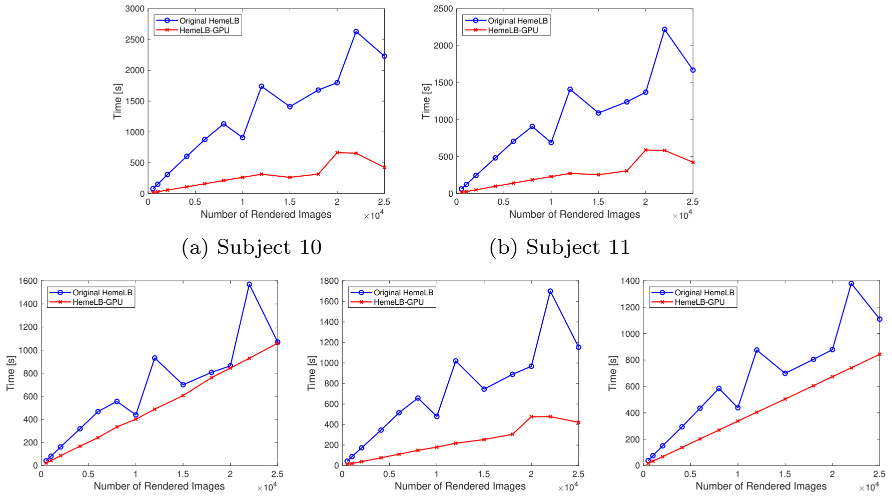
The results in Fig. 11 illustrate that for all the geometries, the HemeLB-GPU version performs better than the original HemeLB. It is worth mentioning that by increasing the number of rendered images within the total number of simulation steps, the proposed visualization client consumes less time in comparison to the original version. The gradient of the blue line (i.e. Original HemeLB version) is higher than the gradient of the red line (i.e. HemeLB-GPU version), which means that the HemeLB-GPU version has better scalability than the original HemeLB in terms of accelerating rendering speed. Furthermore, in this highly scalable and distributable HemeLB simulation framework, the performance of the system will not be significantly affected along with the additional computing resources. This is because the proposed rendering engine is able to visualize part of the block while the simulation is running and does not need to wait for the simulation to be completed.



(a) Subject 10 (b) Subject 11

(c) Subject 16 (d) Subject 19 (e) Subject 23

Fig. 10: Overall performance comparison of the original HemeLB and HemeLB-GPU version. The total simulation steps as well as the number of rendered images are set to 25,000.



(a) Subject 10 (b) Subject 11

(c) Subject 16 (d) Subject 19 (e) Subject 23

Fig. 11: Comparison of the HemeLB-GPU version vs. original HemeLB in terms of rendering speed with 8 cores.

In order to investigate how size of a model can affect the performance of the system, an overview of the simulation domain and timing performance for different subjects are represented in Table 2. In this simulation environment, each subject consists of a number of lattice sites (fluid and solid) and a group of lattice sites is called a block. In Table 2, SUPS/Sites depicts the number of times a model is updated per second. The results show that for the subjects with higher number of blocks, more time is spent on communication and therefore, the model update rate will decrease.

Table 2: Update rate and timing performance of the proposed system.

Subject	Number of Blocks	Number of Sites	Total Time (s)	SUPS	SUPS/Sites
Subject 19	728	81,952	421	4,866,508	59.38242
Subject 11	2,520	151,700	423	8,965,721	59.10165
Subject 10	2,880	186,463	424	10,994,281	58.96227
Subject 23	5,320	132,910	845	3,932,249	29.58580
Subject 16	15,000	296,814	1,060	7,000,330	23.58491

The 3D visualization of results has also been achieved with more than 30 Frames Per Second (FPS), which indicates that the proposed system meets real-time data processing requirements. The feedback from the clinicians shows that this fast visual and interactive simulation platform would provide them a convenient tool to view the subjects from different angles and check the results. They were mainly impressed with zooming the 3D volumes in and out, the blood simulation color gradient and the simple interaction with the framework. On the other hand, the real-time visualization would reduce the preparation time for clinicians to achieve early stage diagnosis.

Finally, the results show that having more lattice sites and blocks gives better resolution and outputs better visualization result, which means it may give more clinical information to the users. Because the HemeLB has its own tool to customize the resolution of the simulation, an optimised result could be given based on the clinical requirements.

6 Conclusion

In this paper, we have presented an interactive cerebral aneurysm blood flow simulation and real-time visualization pipeline. The proposed pipeline addresses all the components required for blood flow modelling and investigation. An improved version of HemeLB has been deployed in this pipeline as a fluid solver which is optimized for visualization on GPU cores. This real-time GPU based rendering engine enables HemeLB to be applied in hospitals and helps clinicians to investigate blood flow into and around cerebral aneurysms in a clinical time-frame. In addition, a visualization platform with the aim of providing an interactive environment for the user is also designed. This platform gives easy access to different components of the pipeline by a few simple clicks. Finally, a set of comprehensive tests using real patients' data has been carried out to evaluate the rendering performance of the pipeline. The reported results demonstrate that this rendering engine performs significantly better than the original HemeLB. On average, the image rendering time in the proposed version is less than half of that in the original version. This ensures that the presented pipeline can be employed in local environments to explore the blood flow patterns in vivo.

As future work, we are going to add simulation steering capabilities to the pipeline. This can help the user to modify the flow properties dynamically and examine the results of moving a stent around inside an aneurysm in real-time. Additionally, a new multi-core architecture for the visualization client can also be developed in order to improve the timing performance. This can be achieved by giving all the nodes access to the GPU memory and hence eliminating data transfers from LB nodes to the master node.

Acknowledgements This study was made possible by a National Priorities Research Program (NPRP) grant No. 5-792-2-328 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. This article does not contain any studies with animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. L. Pierot, L. Spelle, X. Leclerc, C. Cognard, A. Bonafé, J. Moret, Endovascular treatment of unruptured intracranial aneurysms: comparison of safety of remodeling technique and standard treatment with coils, Radiology **251**(3), 846 (2009)
2. T. NeuroDocs. Rupture rates for small brain aneurysms. https://www.topneurodocs.com/rupture-rates-small-brain-aneurysms (2010). Accessed 2019
3. M.D. Mazzeo, 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 (2008)
4. X. Zhai, A. Amira, F. Bensaali, A. Al-Shibani, A. Al-Nassr, A. El-Sayed, M. Eslami, S.P. Dakua, J. Abinahed, Zynq SoC based acceleration of the lattice Boltzmann method, Concurrency and Computation: Practice and Experience p. e5184 (2019)
5. F. Kuznik, C. Obrecht, G. Rusaouen, J.J. Roux, Lbm based flow simulation using gpu computing processor, Computers & Mathematics with Applications **59**(7), 2380 (2010)
6. H. Djelouat, X. Zhai, M. Al Disi, A. Amira, F. Bensaali, System-on-Chip solution for patients biometric: A compressive sensing-based approach, IEEE Sensors Journal **18**(23), 9629 (2018)

7. D. Groen, R.A. Richardson, R. Coy, U.D. Schiller, H. Chandrashekar, F. Robertson, P.V. Coveney, Validation of patient-specific cerebral blood flow simulation using transcranial Doppler measurements, Frontiers in Physiology **9**, 721 (2018)
8. A. Patronis, R.A. Richardson, S. Schmieschek, B.J. Wylie, R.W. Nash, P.V. Coveney, Modelling Patient-Specific Magnetic Drug Targeting within the Intracranial Vasculature, Frontiers in physiology **9**, 331 (2018)
9. D. Groen, J. Hetherington, H.B. Carver, R.W. Nash, M.O. Bernabeu, P.V. Coveney, Analysing and modelling the performance of the HemeLB lattice-Boltzmann simulation environment, Journal of Computational Science **4**(5), 412 (2013). DOI 10.1016/j.jocs.2013.03.002
10. H. Shi, J. Chen, W. Pan, K.S. Hwang, Y.Y. Cho, Collision avoidance for redundant robots in position based visual servoing, IEEE Systems Journal **1**(99), 1 (2018)
11. X. Zhai, M. Chen, S. Soheilian, A. Amira, F. Bensaali, J. AbiNahed, S. Daku, A. Al-Ansari, A. Zakaria, *Zynq SoC Based Lattice-Boltzmann Simulation Environment*, in *10th IEEE GCC Conference and Exhibition: Powering the 4th Industrial Revolution* (IEEE, 2019)
12. M.D. Mazzeo, S. Manos, P.V. Coveney, In situ ray tracing and computational steering for interactive blood flow simulation, Computer Physics Communications **181**(2), 355 (2010)
13. T. Kalaiselvi, P. Sriramakrishnan, K. Somasundaram, Survey of using GPU CUDA programming model in medical image analysis, Informatics in Medicine Unlocked **9**, 133 (2017)
14. G. Ramesh Chandra, E. Rajan, Improving the Performance of Volume Rendering for Medical Images, International Journal of Advances in Soft Computing Technology **1**, 2229 (2013)
15. R.F. McCormack, A. Hutson, Can computed tomography angiography of the brain replace lumbar puncture in the evaluation of acute-onset headache after a negative noncontrast cranial computed tomography scan?, Academic Emergency Medicine **17**(4), 444 (2010). DOI 10.1111/j.1553-2712.2010.00694.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1553-2712.2010.00694.x
16. E. Uysal, B. Yanbuloglu, M. Erturk, B. Kilinc, M. Basak, Spiral CT angiography in diagnosis of cerebral aneurysms of cases with acute subarachnoid hemorrhage, Diagn Interv Radiol **11**, 77 (2005)
17. A. McKinney, C. Palmer, C. Truwit, A. Karagulle, M. Teksam, Detection of aneurysms by 64-section multidetector ct angiography in patients acutely suspected of having an intracranial aneurysm and comparison with digital subtraction and 3d rotational angiography, American Journal of Neuroradiology **29**(3), 594 (2008). DOI 10.3174/ajnr.A0848. URL http://www.ajnr.org/content/29/3/594
18. K. Nael, J. Villablanca, R. Saleh, W. Pope, A. Nael, G. Laub, J. Finn, Contrast-enhanced mr angiography at 3t in the evaluation of intracranial aneurysms: A comparison with time-of-flight mr angiography, American Journal of Neuroradiology **27**(10), 2118 (2006). URL http://www.ajnr.org/content/27/10/2118
19. P.D. Schellinger, G. Richter, M. Kohrmann, A. Dorfler, Noninvasive Angiography (Magnetic Resonance and Computed Tomography) in the Diagnosis of Ischemic Cerebrovascular Disease, Radiology **24(suppl 1)**(Suppl. 1), 16 (2007)
20. M. Bernstein, J. Huston, C. Lin, G. Gibbs, J. Felmlee, High-resolution intracranial and cervical mra at 3.0t: Technical considerations and initial experience, Magnetic Resonance in Medicine **46**(5), 955 (2001). DOI 10.1002/mrm.1282
21. J. Forget, Thomas R., R. Benitez, E. Veznedaroglu, A. Sharan, W. Mitchell, M. Silva, R.H. Rosenwasser, A Review of Size and Location of Ruptured Intracranial Aneurysms, Neurosurgery **49**(6), 1322 (2001). DOI 10.1097/00006123-200112000-00006. URL https://doi.org/10.1097/00006123-200112000-00006
22. W.J. van Rooij, M. Sprengers, A.N. de Gast, J. Peluso, M. Sluzewski, 3D rotational angiography: the new gold standard in the detection of additional intracranial aneurysms, American Journal of Neuroradiology **29**(5), 976 (2008)
23. S.P. Dakua, J. Abinahed, A. Al-Ansari, P.G. Bermejo, A. Zakaria, A. Amira, F. Bensaali, *A Method Towards Cerebral Aneurysm Detection in Clinical Settings*, in *Sipaim–Miccai Biomedical Workshop* (Springer, 2018), pp. 8–15
24. X. Zhai, M. Eslami, E.S. Hussein, M.S. Filali, S.T. Shalaby, A. Amira, F. Bensaali, S. Dakua, J. Abinahed, A. Al-Ansari, A. Z. Ahmed, Real-time automated image segmentation technique for cerebral aneurysm on reconfigurable system-on-chip, Journal of computational science **27**, 35 (2018)

25. M.A. Itani, U.D. Schiller, S. Schmieschek, J. Hetherington, M.O. Bernabeu, H. Chandrashekar, F. Robertson, P.V. Coveney, D. Groen, An automated multiscale ensemble simulation approach for vascular blood flow, Journal of Computational Science **9**, 150 (2015). DOI 10.1016/j.jocs.2015.04.008
26. M. Bouzidi, M. Firdaouss, P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, Physics of fluids **13**(11), 3452 (2001)
27. A.J. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation, Journal of fluid mechanics **271**, 285 (1994)
28. J.K. Udupa, H.M. Hung, K.S. Chuang, Surface and volume rendering in three-dimensional imaging: a comparison, Journal of digital Imaging **4**(3), 159 (1991)
29. K. Zhou, Z. Ren, S. Lin, H. Bao, B. Guo, H.Y. Shum, *Real-time smoke rendering using compensated ray marching*, in *ACM Transactions on Graphics (TOG)*, vol. 27 (ACM, 2008), vol. 27, p. 36
30. S.S. Esfahani, X. Zhai, M. Chen, A. Amira, F. Bensaali, J. AbiNahed, S. Dakua, G. Younes, R. A. Richardson, P. V. Coveney, *HemeLB acceleration and visualization for cerebral aneurysms*, in *IEEE International Conference on Image Processing (ICIP)* (IEEE, 2019)
31. E. Corporation. Exxact Tensor TWS-289059-DPN specification. https://www.exxactcorp.com/Exxact-TWS-289059-DPN-E289059 (2018). Accessed 25 December 2018
32. Moai. Moai SLA printer and accessories. https://peopoly.net/collections/frontpage (2018). Accessed 2018
33. Olympus. Peristaltic Pump. http://items.olympus.eu/common/epaper-oste/index.html (2018). Accessed 2018
34. Philips. Verrata Pressure guide wire. https://www.usa.philips.com/healthcare/product/HC989604186581/verrata-pressure-guide-wire (2018). Accessed 2018
35. Medtronic. Pipeline Flex Embolization Device. https://www.medtronic.com/us-en/healthcare-professionals/products/neurological/hemorrhagic-stroke/pipeline-flex.html (2018). Accessed 2018