



Research article

A GPU accelerated study of aqueous humor dynamics in human eyes using the lattice Boltzmann method

Gang Huang^{1,2}, Qianlin Ye^{1,*}, Hao Tang² and Zhangrong Qin^{2,*}

¹ Business School, Jiangxi University of Science and Technology, Jiangxi, China

² Guangxi Key Lab of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China

* **Correspondence:** Email: jxlgyql@163.com, qinzhangrong@gxnu.edu.cn.

Abstract: In this paper, we presented a 3D human eyes aqueous humor (AH) dynamics model, and additionally, designed and optimized it using GPU technology. First, the feasibility of the model is demonstrated through validation. Then, the effect of different factors on AH flow was investigated using the validated model. The experimental results showed that AH flow more rapidly when standing than supine; the intraocular temperature has the greatest effect on AH flow compared to other factors; the AH secretion rate and trabecular meshwork (TM) permeability had a greater effect on intraocular pressure (IOP). Corneal indentation and ovoid anterior chamber (AC) can also affect AH flow. Finally, the PartSparse algorithm based GPU can save more than 50% of the memory consumption and achieves a performance of 1491.29 MLUPS and a Speedup of 837.61 times.

Keywords: lattice Boltzmann method; aqueous humor dynamics; GPU; sparse matrix

1. Introduction

Ophthalmic diseases are diverse and have complex causes, and studies have shown a strong association between AH dynamics and ophthalmic diseases [1–3]. Therefore, it is very necessary to study the laws of AH dynamics in human eyes, which can improve the understanding of the pathogenesis and development of ophthalmic diseases, as well as the effectiveness of drug therapy.

In recent years, many scholars have conducted numerous studies on AH dynamics using numerical methods. Canning et al. [1] proposed a simple model of AH thermal convection based on

the Boussinesq approximation. He found that natural convection occurs in the AC, even at very small temperature differences, and concluded that temperature difference is the most important factor driving AH flow, but Canning did not consider the secretion-discharge mechanism of AH. Heys and Barocas [2] constructed a three-dimensional (3D) model of AH flow in the human eyes by Boussinesq approximation and studied the flow of AH during standing at different intraocular temperature differences. He also found that the secretion-discharge of AH had little effect on the AH flow, but did not consider the AH flow in the supine position. Fitt and Gonzalez [3] investigated the different factors causing AH flow and further analyzed the effects of buoyancy forces (temperature difference assumed to be 0.1°C) and rapid eye movements, as well as lens tremor on AH flow during sleep, in addition to both natural convection and secretory-discharge flow. Crowder and Ervin [4] explored the effect of different AH secretion rates on AH flow and showed that the pressure difference between the AC and the TM was greater with higher secretion rates, but only the supine position was considered. Kumar et al. [5] developed a physical model from rabbit eye anatomical data and incorporated a TM with porous media properties and applied Darcy's law to calculate the AH velocity at the TM. He used FLUENT to solve the problem, and found that the buoyancy force was the main driving force for AH flow after comparing the effect of buoyancy force and AH secretion-discharge. Ferreira et al. [6] studied the effects of drugs on the ciliary body (CB) and TM, and first determined that higher AH secretion rates or smaller TM permeability were more likely to result in high IOP, which could be significantly reduced by drug treatment.

In summary, researchers have conducted a large number of studies on the AH dynamics of the human eyes and have obtained many results, that have laid a solid foundation for an in-depth study of AH dynamics. However, much is still unknown about AH dynamics, such as: the flow of AH at different eye orientations and the effect of asymmetric AC on AH flow. Moreover, most of the current research uses traditional numerical methods such as finite element or finite difference, which have shortcomings, such as not easy to deal with complex boundaries and difficult to compute in parallel.

The Lattice Boltzmann Method (LBM) is a mesoscopic numerical method between macroscopic and microscopic, which originated from lattice gas automata. LBM treats fluid as a lattice distributed on a regular grid, following correlation laws for collision and migration (flow), and then uses statistical averaging to calculate the macroscopic physical quantities (e.g., density, velocity, pressure, temperature, etc.). The original lattice Boltzmann (LB) model was proposed by McNamara and Zanetti [7], who introduced a local particle distribution function to represent the number of particles per lattice. Then, Chen et al. [8] and Qian et al. [9] proposed a simplified LB model by replacing the equilibrium state distribution function with the local particle distribution function and linearizing the collision operator, and the two's method is known as the lattice BGK (LBGK) model. Further, with the extensive development of LBM, many models have been derived, such as the thermal lattice model [10] and the double distribution function model [11,12]. Due to the advantages of easy multi-physical coupling, simple boundary processing, and easy parallelism, LBM has been successfully applied in complex fluid fields, such as heat transfer [11], and porous media flow [12]. In the field of AH flow, Qin et al. [13] established the LB model of AH dynamics with two-dimensional flow field, which is based on the incompressible fluid model and better simulates the AH flow phenomenon. Additionally, he investigates factors, such as IOP and AH secretion rate. Due to the limitation of the model, the three-dimensional flow field cannot be simulated.

In LBM, lattice points need to collide and flow with neighboring lattice points frequently during

evolution, especially in the 3D flow field, which involves a large amount of computation, and, thus, there is an urgent need to improve the computational efficiency. The GPU with “Single Instruction Multiple Thread” (SIMT) [14] parallel mode is well suited for LBM with natural parallelism features, which can significantly improve the performance of LBM algorithms [15–22].

The study of 3D human eyes AH dynamics involves the complex anatomical structure of the human eyes, difficult boundary processing, and huge computation. Therefore, it is a promising new way to study the AH dynamics of the human eyes by using the LBM, which has the advantages of easy multi-physical coupling, simple boundary processing, and easy parallelism. The parallel design and optimization of 3D human eyes AH dynamics model by GPU technology can effectively improve the computational performance of the model and shorten the research cycle. The flow laws of AH in human eyes in three dimension can be obtained through this study, which is expected to provide some theoretical references for clinical diagnosis and treatment.

This paper is structured as follows: Section 2 describes the modeling process of the coupled AH dynamics LB model, including the geometric model of the anterior segment, the macroscopic equations of AH flow and the corresponding LB model with double distribution function. Section 3 uses GPU technology for parallel design and optimization of the 3D human eyes AH dynamics model. In Section 4, the 3D AH dynamics model is first validated, and then the effects of intraocular temperature difference, AH secretion rate, TM permeability, corneal indentation and asymmetric anterior chamber cavity on AH flow are investigated. Finally, we compare the impact of GPU parallel algorithms on program performance.

2. LB model of AH dynamics in the human eyes

AH dynamics occurs primarily in the anterior segment. AH is secreted by the CB located in the posterior chamber (PC) at a rate of 1.5~3.0 $\mu\text{l}/\text{min}$, flows through the narrow PC, then across the pupil into the AC, where it forms a circulating convection driven by temperature difference, and is finally discharged through the TM [23]. The flow of AH is very complex and characterized by multi-scale and multi-physical coupling, making it suitable for study using LBM.

2.1. Geometry

We developed an idealized 3D geometric model of the anterior segment through the anatomical structure of the human eyes. The anatomical structure of the anterior segment is shown in Figure 1(a) and (b), and its main ocular tissues are: Cornea, Iris, Lens, Pupil, CB, TM and Suspensory ligament. The area between the cornea, iris and pupil is called the AC, and the area enclosed by the iris, lens and suspensory ligament is called the PC. The AC is filled with an aqueous fluid called AH, which is 98% water and has physical properties very similar to water, and its main role is to transport nutrients and remove metabolic wastes for the tissues of the eye.

The geometric model of the three-dimensional anterior segment is shown in (a) and (b) of Figure 1, and the corresponding sizes of each tissue are shown in Table 1. The geometry is 12.8 mm \times 12.8 mm \times 4.0 mm, and the computational grid is 384 \times 384 \times 120 by dimensional conversion.

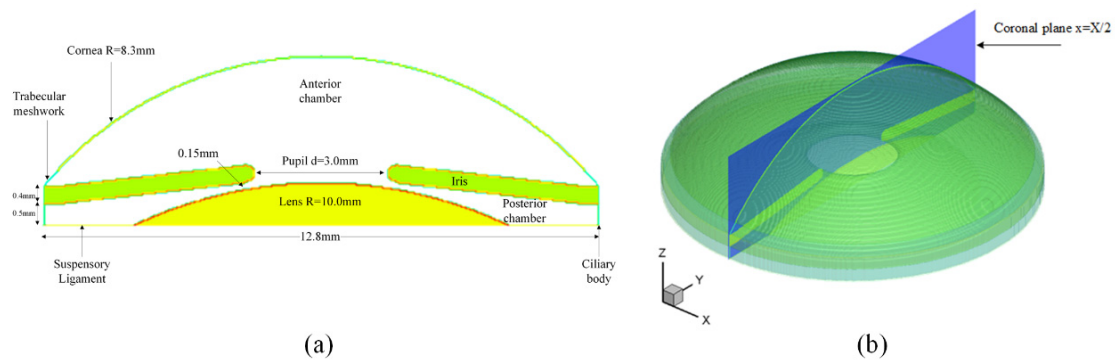


Figure 1. Anterior segment geometric model: (a) Coronal plane, (b) Three-dimensional.

Table 1. Geometric parameters of the tissues.

Geometric parameter	Value	Source
Length of the AC	12.8 mm	Abouali et al. [24]
Depth of the AC	3.0 mm	Ferreira et al. [6]
Radius of the cornea	8.3 mm	Heys and Barocas [2]
Thickness of the iris	0.4 mm	Heys et al. [23]
Angle of the iris	6.45°	Lin and Yuan [25]
Diameter of the pupil	3.0 mm	Abouali et al. [24]
Radius of the lens	10.0 mm	Heussner et al. [26]
Surface area of the TM	18.0 mm ²	Heys et al. [23]
Height of the PC	0.5 mm	Heys et al. [23]
Distance between iris and lens	0.15 mm	Between Heys et al. [23] and Crowder and Ervin [4]

2.2. Governing equations

AH is considered as a Newtonian fluid and flows at a very small velocity with negligible compressibility and inertia. Therefore, the incompressible Navier-Stokes (N-S) equation can be used to describe the flow process of AH within the anterior segment:

$$\nabla \cdot \mathbf{u} = 0 \quad (1a)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \times \mathbf{u} \times \mathbf{u}) = -\nabla p + \nu \nabla^2 \mathbf{u} - \mathbf{F}, \quad (1b)$$

where, \mathbf{u} is the velocity of the AH, p is the pressure of the AH, ν is the kinematic viscosity of the AH, and \mathbf{F} is the external force.

Because of the temperature difference between the corneal surface of the eye and the intraocular tissues, heat convection diffusion within the anterior segment of the eye must be considered. It is well known that if the pressure work and viscous heat dissipation are neglected, the temperature field is subject only to passive advection of the fluid and obeys a simple passive scalar equation:

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = \alpha \nabla^2 T, \quad (2)$$

where T is the temperature and α is the thermal diffusion coefficient. AH is discharged through the TM, a tissue with porous media properties and specific thickness. According to the theory of porous media, the seepage flow of AH at the TM obeys Darcy's law:

$$\mathbf{u} = -\frac{K \Delta p}{\mu L}, \quad (3)$$

where \mathbf{u} is the seepage flow velocity of AH at the TM, K is the permeability, Δp is the pressure difference, μ is the AH viscosity and L is the thickness of the TM.

In AH dynamics, the coexistence of incompressible N-S flow, thermal convective diffusion and Darcy seepage flow constitutes the complex AH flow in the AC. Since the flow of AH in the AC and PC is natural convection, the Boussinesq approximation can be used to couple the incompressible N-S flow with thermal convection diffusion. The Boussinesq approximation assumes that the density, viscosity and thermal diffusivity of the fluid are constant except for buoyancy, which satisfies:

$$\rho = \rho_0 [1 - \beta(T - T_0)], \quad (4)$$

where T_0 is the reference temperature, ρ_0 is the reference density and β is the coefficient of thermal expansion. Under the Boussinesq approximation, gravity can be expressed as:

$$\mathbf{G} = \rho_0 \mathbf{g} - \rho_0 \mathbf{g} \beta (T - T_0), \quad (5)$$

where \mathbf{g} is the gravitational acceleration vector. The constant part of the first term is absorbed into the pressure term to obtain the external force term:

$$\mathbf{F} = -\mathbf{g} \beta (T - T_0). \quad (6)$$

In summary, the set of equations for the flow of AH based on the Boussinesq approximation is:

$$\nabla \cdot \mathbf{u} = 0 \quad (7a)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \times \mathbf{u}) = -\nabla p + \nu \nabla^2 \mathbf{u} - \mathbf{g} \beta (T - T_0) \quad (7b)$$

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = \alpha \nabla^2 T. \quad (7c)$$

Coupling incompressible N-S flow and Darcy flow, the IOP can be calculated by Eq (7b) first, then the pressure difference between AH and vein is found, and the seepage flow velocity is obtained by Darcy equation, and, finally, the seepage flow velocity is used for the velocity boundary, and the coupling of Eqs (7) and (3) is realized by repeating iterations.

2.3. Using LBM to solve macroscopic equations

In the LBM, the discrete velocity distribution of fluid particles in space and time is described by a density distribution function $f_i(\mathbf{x}, t)$. The LBGK model with a single-relaxation-time approximation can be expressed as follows:

$$f_i(\mathbf{x} + \mathbf{e}_i \delta_x, t + \delta_t) - f_i(\mathbf{x}, t) = -\frac{1}{\tau_f} (f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)) + \delta_t F_i, \quad (8)$$

where \mathbf{x} is the spatial position of the particle, t is time, \mathbf{e}_i is the discrete velocity, τ_f is the relaxation time, $f_i(\mathbf{x}, t)$ is the particle distribution function, $f_i^{eq}(\mathbf{x}, t)$ is the corresponding equilibrium distribution function δ_x and δ_t is the spatial and temporal increments, respectively, and F_i is the external force.

In the lattice Boltzmann of AH dynamics, both velocity and temperature fields exist, therefore, the double distribution function LB model based on Boussinesq approximation is used for the solution. Extending the double distribution function LB model of Guo and Zhao [12] to the three-dimensional case, the density equilibrium distribution function is:

$$f_i^{(eq)} = \rho \omega_i \left[1 + \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{|\mathbf{u}|^2}{2c_s^2} \right] \quad (i = 0, 1, 2, \dots, 18), \quad (9)$$

where ρ is the particle density, ω_i is the weight coefficient, \mathbf{u} is the particle velocity, $c_s = c/\sqrt{3}$ and $c = \delta_x/\delta_t$ (value is 1). The external force is:

$$F_i = \left(1 - \frac{1}{2\tau_f} \right) \omega_i \left[\frac{\mathbf{e}_i - \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u})}{c_s^4} \mathbf{e}_i \right] \cdot \mathbf{F} \quad (10)$$

The discrete velocity \mathbf{e} and the weight coefficient ω of the D3Q19 lattice model are:

$$\mathbf{e}_i = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \end{bmatrix} \quad (11)$$

$$\omega_i = \begin{cases} 1/3, & i = 0 \\ 1/18, & i = 1, 2, \dots, 6 \\ 1/36, & i = 7, 8, \dots, 18 \end{cases} \quad (12)$$

The evolution equation of the temperature field:

$$g_i(\mathbf{x} + \mathbf{e}_i \delta_x, t + \delta_t) - g_i(\mathbf{x}, t) = -\frac{1}{\tau_g} (g_i(\mathbf{x}, t) - g_i^{eq}(\mathbf{x}, t)) \quad (i = 0, 1, 2, \dots, 6), \quad (13)$$

where $g_i(\mathbf{x}, t)$ is the temperature distribution function of the particle, $g_i^{eq}(\mathbf{x}, t)$ is the corresponding equilibrium distribution function and τ_g is the relaxation time of the temperature field.

Equilibrium distribution function of temperature:

$$g_i^{eq} = \eta_i T \left[1 + \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} \right] \quad (i = 0, 1, 2, 3, 4), \quad (14)$$

where, T is the particle temperature and η_i is the weight coefficient. The discrete velocity \mathbf{e} and the weight coefficient η_i of the D3Q7 lattice model are:

$$\mathbf{e}_i = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad (15)$$

$$\eta_i = \begin{cases} 1/4, & i = 0 \\ 1/8, & i = 1, 2, \dots, 6 \end{cases} \quad (16)$$

The corresponding macroscopic densities, temperatures and velocities are defined as follows:

$$\rho = \sum_{i=0}^{18} f_i \quad (17)$$

$$T = \sum_{i=0}^6 g_i \quad (18)$$

$$\rho \mathbf{u} = \sum_{i=0}^{18} \mathbf{e}_i f_i + \frac{\delta_t}{2} \mathbf{F} \quad (19)$$

The relaxation time is satisfied:

$$\nu = \left(\tau_f - \frac{1}{2} \right) c_s^2 \delta_t \quad (20)$$

$$\alpha = \left(\tau_g - \frac{1}{2} \right) c_s^2 \delta_t \quad (21)$$

In AH dynamics, the external force is provided by the buoyancy force:

$$\mathbf{F} = -\mathbf{g}\beta(T - T_0) \quad (22)$$

In the velocity field, a half-way bounce scheme is used at all solid wall boundaries and a non-equilibrium extrapolation scheme is used at the CB and TM [27], while the temperature field uses a non-equilibrium extrapolation scheme at all boundaries.

3. GPU-based LBM parallel algorithm

Since NVIDIA introduced the CUDA (Compute Unified Device Architecture) programming technology on GPUs, heterogeneous CPU and GPU-based parallel acceleration models have become increasingly popular due to their lower cost and ease of use. Moreover, the GPU in parallel computing mode is well suited for LBM, and can significantly improve the performance of LBM

algorithms [15–22], making it suitable for studying AH dynamics. However, GPU-based LBM algorithms require frequent reads from global memory, so the way in which computational data is laid out in global memory is particularly important. It has been demonstrated that the SOA model of memory layout satisfies merge access and is more suitable for GPU-based LBM algorithms [20,28–30].

In our algorithm, the SOA model [20] is used to allocate memory for the particle distribution function. Meanwhile, based on the characteristics of CPU and GPU, the steps of initializing variables, establishing flow fields and outputting data are implemented on the CPU, while the steps of collision, flow (including boundary processing) and calculating macroscopic quantities of the evolutionary process are implemented on the GPU. Further, to improve performance and reduce thread access to global memory, collision and flow are combined in a single core function [20].

The flow field of the D3Q19 lattice model requires three-dimensional array memory, and for ease of manipulation, we use a one-dimensional array to represent the three-dimensional flow field. The flow field size of the 3D AH dynamics model is $DX = 384$, $DY = 384$ and $DZ = 120$, and the total number of flow field lattices is $DN = DX \times DY \times DZ$. Allocate memory to lattices in the order of Z, Y, X and convert them to the form of a one-dimensional array. After conversion, the lattice index is $p = I \times DY \times DZ + j \times DZ + k$ (i, j, k are the coordinates in the X, Y, and Z directions, respectively), while the corresponding distribution function is $pf = f \times DX \times DY \times DZ + p$ [31].

In the flow field, the fluid and solids are integrated with each other, and the fluid lattices account for about 38% of the total lattices in the flow field, so the 3D AH dynamics flow field has geometrically sparse characteristics. Currently, there are fewer solutions for complex flow fields, especially those with geometrically sparse properties, such as indirect addressing [29] and sparse matrix [32] methods. Additionally, this paper proposes a PartSparse scheme based on sparse matrices, which can further improve the computational performance while reducing the program memory consumption.

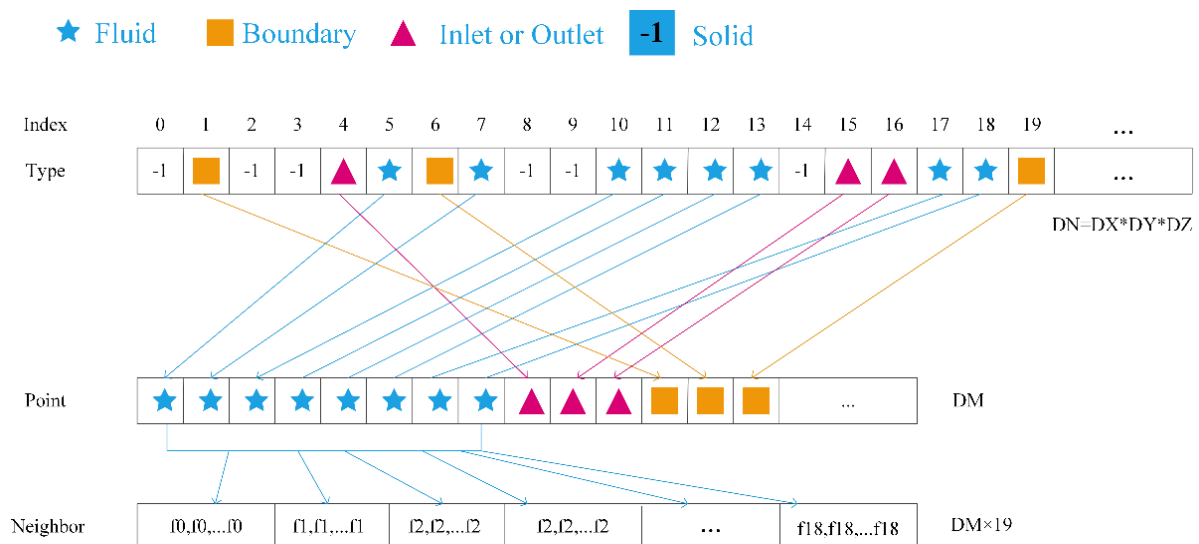


Figure 2. PartSparse scheme.

The idea of the PartSparse scheme is shown in Figure 2. First, the 3D AH dynamics flow field is converted into a one-dimensional array Type with the total number of flow field lattices DN ($DN = DX \times DY \times DZ$). Then, based on the sparse property of the flow field, the solid lattices not involved

in the evolution are removed, and the lattices are sorted in parts by type (fluid lattices at the head, entrance and exit boundary lattices at the middle, and solid wall boundary lattices at the tail) to obtain the lattice array Point, whose size is DM and the corresponding lattice index is p. Finally, according to the rules of the D3Q19 lattice model, collisions and flows are needed 19 times during each iteration, using Neighbor to represent an array of neighbors of lattices, whose size is $DM \times 19$, and the index of elements is $pf = p + f*DM$ (f denotes 0, 1, 2,, 18 directions). Thus, in order to meet the memory merge access, Neighbor's memory layout also adopts the SOA model, i.e., it is stored according to the flow direction of lattices. The pseudo-code for the Initialization and Evolution of the PartSparse scheme is shown in Algorithm 1 and Algorithm 2, where N_Fluid, N_InletOutlet and N_Boundary represent the total number of lattices in the flow field at the fluid, entrance/exit boundaries and Boundary, respectively.

Algorithm 1 Initialization of PartSparse

```

1: N_Fluid, N_InletOutlet, N_Boundary, Index[], Type[], Point[], Neighbor[]
2: d_Fluid = d_InletOutlet = d_Boundary = 0
3: for i = 0 → DX do for j = 0 → DY do for k = 0 → DZ do
4:   p = i*DY*DZ + j*DZ + k, Index[p] = -1
5:   if Type[p] == Fluid then
6:     int n = d_Fluid++, Index[p] = n
7:     use Point[n] to set Fluid parameters
8:   end if
9:   else if Type[p] == InletOutlet then
10:    int n = N_Fluid + d_InletOutlet, Index[p] = n, d_InletOutlet++
11:    use Point[n] to set InletOutlet parameters
12:   end if
13:   else if Type[p] == Boundary then
14:    int n = N_Fluid + N_InletOutlet + d_Boundary, Index[p] = n, d_Boundary++
15:    use Point[n] to set Boundary parameters
16:   end if
17: end for
18:// set neighbor points
19: for i = 0 → DX do for j = 0 → DY do for k = 0 → DZ do
20:   p = i*DY*DZ + j*DZ + k
21:   if Index[p] < 0 continue
22:   end if
23:   for f = 0 → 18 do
24:     calculate ii, jj and kk using i, j, k and Eq (11)
25:     int pp = ii*DY*DZ + jj*DZ + kk
26:     int nf = n + f*DM
27:   end for
28: end for

```

Algorithm 2 Evolution of PartSparse

```

1: N_Fluid, N_InletOutlet, Neighbor[]
2: p = blockIdx.x*blockDim.x+threadIdx.x
3: if p >= N_Fluid    return
4: for f = 0 → 18 do
5:     pf = p + f*DM
6:     pp = Neighbor[pf]
7:     ppf = pp + f*DM
8:     calculate  $F_i$  using Eq (10)
9:     if pp < N_Fluid then
10:        calculate  $f_i$  using Eq (8)
11:        if f < 7 then
12:            calculate  $g_i$  using Eq (13)
13:        end if
14:    else if pp >= N_Fluid && pp < (N_Fluid + N_InletOutlet) then
15:        calculate Inlet and Outlet
16:    else
17:        calculate Boundary
18:    end if
19: end for
20: end if

```

4. Simulation of AH dynamics

In this section, we simulate the AH flow using a GPU-accelerated 3D human eyes AH dynamics model. First, the velocity and temperature distribution of AH in the AC were compared for different eye orientations, and then the effects of different intraocular temperature differences, AH secretion rate and TM permeability on AH flow were investigated. Finally, the effects of corneal indentation and ovoid AC cavity on AH flow were investigated using the current model.

Table 2. Physical properties of AH.

Physical property	Value	Source
AH viscosity, μ	6.947×10^{-4} Pa·s	(water)
AH density, ρ	9.93×10^2 kg/m ³	(water)
AH thermal conductivity, α	0.58 W/(m·K)	Karampatzakis and Samaras [33]
AH specific heat, C_p	4.178×10^3 J/(kg·K)	Zhao et al. [34]
Volume expansion coefficient, β	3.2×10^4 K ⁻¹	(water)
Gravitational acceleration, g	9.81 m/s ²	Zhao et al. [34]

The physical properties of the AH are shown in Table 2. AH is secreted by the CB located in the PC at a rate of 1.5~3.0 $\mu\text{l}/\text{min}$ [23], and in this paper, the AH secretion rate was assumed to be 2.5 $\mu\text{l}/\text{min}$, and the TM permeability was assumed to be $K = 7.0 \times 10^{-15} \text{ m}^2$ [6]. The corneal temperature was set to $T_c = 34^\circ\text{C}$, the rest of the wall temperature was set to $T_h = 37^\circ\text{C}$, and the AH initial temperature was set to $T_0 = (T_h + T_c)/2 = 35.5^\circ\text{C}$. Pr is 5.03, τ_f is 1.0, τ_g can be calculated by $Pr = \nu / \alpha$ and Eq (21).

4.1. Model validation

In this section, we validated the feasibility of the model. We calculated the temperature and velocity distributions of normal AH flow and compared them with the relevant literature. The results are shown in Figure 3, where (a) is the result of this paper, (b) and (c) are the results of Heys and Barocas [2] and Zhao et al. [34], respectively. As can be seen from the figure, the temperature and velocity distribution in this paper are in good agreement with the results in the literature, and the maximum AH velocity is 0.743 mm/s, which is close to the calculated results of Heys (0.7 mm/s) and Zhao (0.684 mm/s).

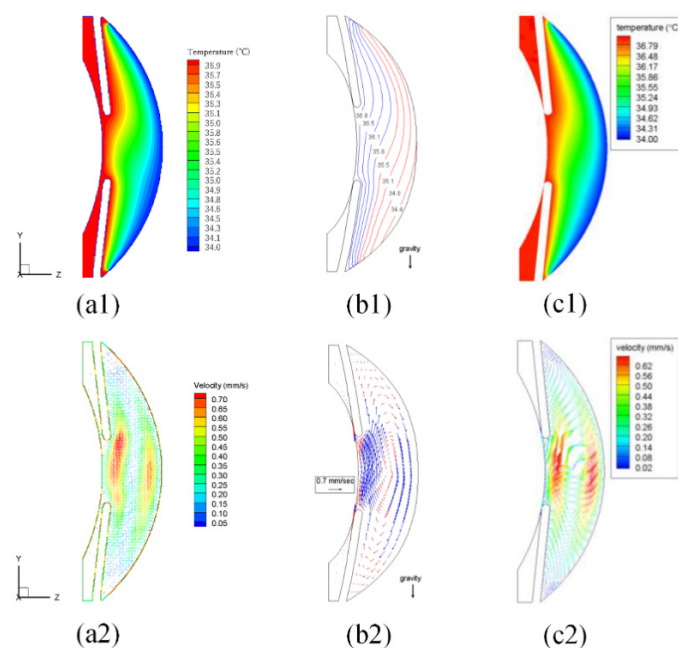


Figure 3. Temperature and velocity distribution of AH: (a) Present, (b) Heys and Barocas [2], (c) Zhao et al. [34].

Then, we simulated the 3D distribution of AH temperature and velocity. Figure 4 gives the temperature and velocity distributions for both stand (gravity along the negative direction of the Y-axis) and supine (gravity along the negative direction of the Z-axis) cases, respectively. As can be seen from the figure, the distribution of temperature and velocity varies greatly at different eye orientations. First, under standing, the AH near the iris is high in temperature and low in density, and rises under gravity. In contrast, the AH near the cornea is low in temperature and high in density, and falls by gravity, resulting in an asymmetric temperature distribution and a clockwise vortex flowing from the

iris to the cornea in the AC. In supine position, the iris temperature is high while the corneal temperature is low, so the AH flows upward from the iris toward the cornea and descends in the direction of the corneal arc. Under gravity, the temperature distribution of AH is symmetrically stepped, and the velocity distribution is centered on the pupil axis forming two symmetrical, oppositely oriented vortices. Finally, the maximum velocity of AH under standing and supine was calculated to be 0.743 and 0.119 mm/s, respectively.

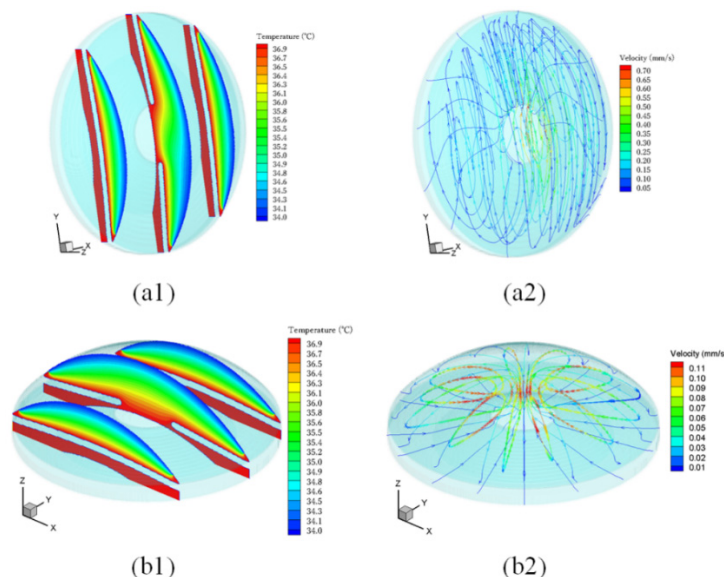


Figure 4. 3D distribution of AH temperature and velocity: (a) Stand, (b) Supine.

4.2. Effect of intraocular temperature difference

Natural convection is the main mode of AH flow, and temperature difference is the key factor leading to natural convection, so the effect of intraocular temperature difference on AH flow is particularly important. In this section, the effect of temperature difference on the dynamics behavior of AH, such as temperature and velocity distribution and flow pattern is investigated by fixing the iris and lens temperature at 37°C, and varying the corneal temperature from 33 to 37°C (temperature difference 0–4°C).

3D graphs are inconvenient to compare in detail, so we compare the temperature and velocity distributions on the coronal plane. The temperature and velocity distributions of the AH for temperature differences of 0, 0.1, 1.0, 2 and 4°C (corresponding to corneal temperatures of 37, 36.9, 36.0, 35 and 33°C) in standing and supine positions are given in Figures 5 and 6, respectively. As can be seen from the figure, as the temperature difference increases, the stronger the temperature diffusion, the faster the AH flow velocity, and the larger the corresponding vortex. It is noteworthy that, when the temperature difference is 0 and 0.1°C, although the temperature diffusion trend is almost the same, the velocity distribution is significantly different. Among them, the temperature difference equal to 0 indicates that the AH is not driven by the buoyancy force and no vortex will be formed. While the temperature difference of 0.1°C, the AH is driven by the buoyancy force, will also form a vortex.

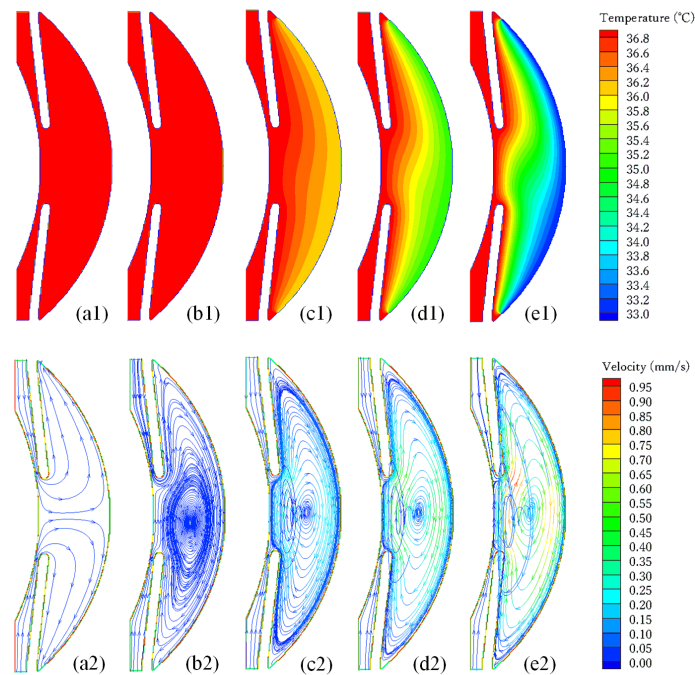


Figure 5. Effect of intraocular temperature difference under standing: (a) $\Delta T = 0^\circ\text{C}$; (b) $\Delta T = 0.1^\circ\text{C}$; (c) $\Delta T = 1.0^\circ\text{C}$; (d) $\Delta T = 2.0^\circ\text{C}$; (e) $\Delta T = 4.0^\circ\text{C}$.

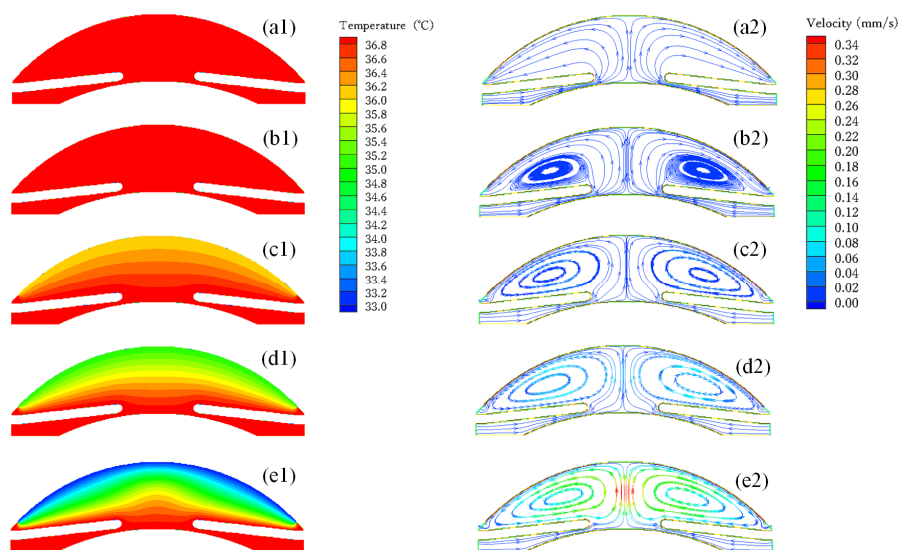


Figure 6. Effect of intraocular temperature difference under supine: (a) $\Delta T = 0^\circ\text{C}$; (b) $\Delta T = 0.1^\circ\text{C}$; (c) $\Delta T = 1.0^\circ\text{C}$; (d) $\Delta T = 2.0^\circ\text{C}$; (e) $\Delta T = 4.0^\circ\text{C}$.

The effect of temperature difference on the maximum velocity of AH is given in Figure 7. As can be seen from the figure, the maximum velocity of AH is almost linearly related to the temperature difference when standing. Under supine, the maximum velocity of AH is more affected by the temperature difference when it is greater than 3°C .

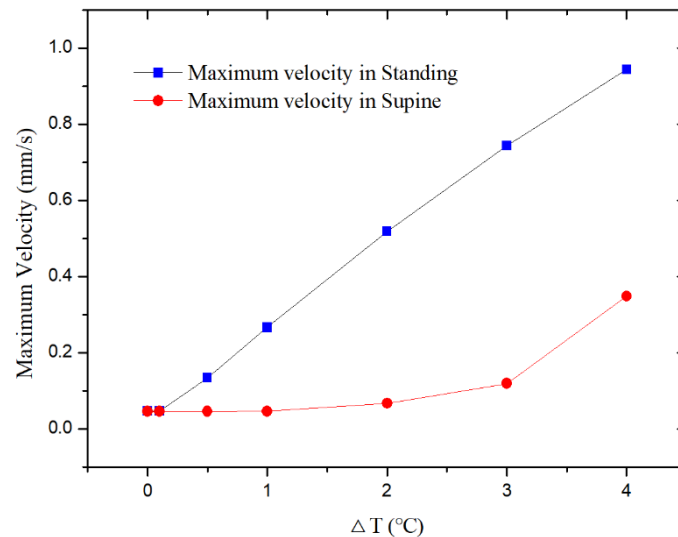


Figure 7. Effect of temperature difference on maximum velocity of AH.

4.3. Effect of IOP

IOP is an important indicator of normal AH flow, and abnormal IOP can lead to ocular disease. In the AH flow system, AH secretion rate and TM permeability are the main effects on AH inflow and discharge, both of which affect the balance of IOP. Therefore, this section simulated the effects of AH secretion rate and TM permeability on IOP. First, normal temperature difference and TM permeability were maintained to study the effect of different AH secretion rates on IOP. Then, the normal temperature difference and AH secretion rate were maintained to study the effect of different TM permeability on IOP.

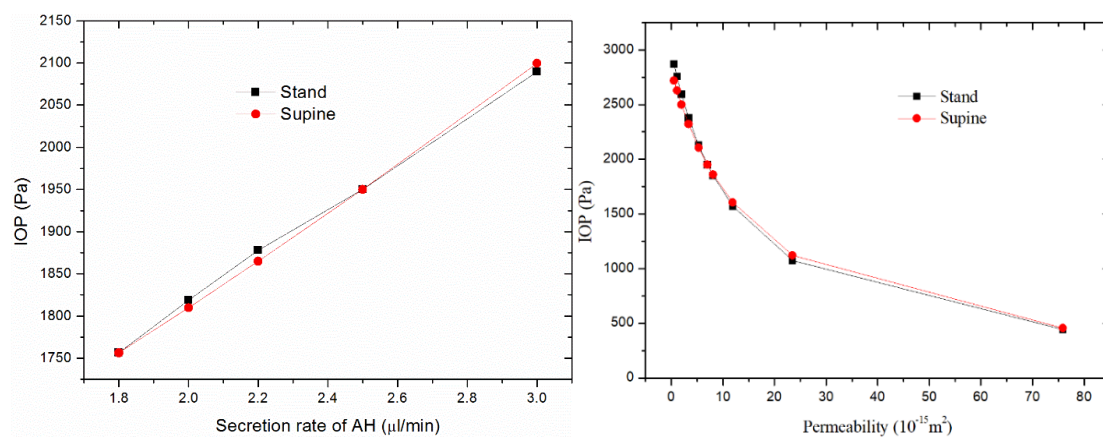


Figure 8. Effect of AH secretion rate and TM permeability on IOP.

The effects of AH secretion rate and TM permeability on IOP are given in Figure 8. As seen in the figure, IOP was significantly affected by AH secretion rate and TM permeability, which were positively proportional to AH secretion rate and inversely proportional to TM permeability. This is because the greater the AH secretion rate, the more AH builds up in the AC, which results in a high

IOP. Additionally, the greater the TM permeability, the more AH flows out, which eventually leads to lower IOP. In summary, normal AH secretion rate and TM permeability are essential to maintain IOP homeostasis.

4.4. Effect of corneal indentation

The cornea can become shaped when it is squeezed by external forces, also known as corneal indentation. Corneal indentation will change the structure of the AC, which in turn will affect the flow of AH. Therefore, to investigate the effect of corneal indentation on AH flow, we simulated the indentation depth in the range of 0.3 mm (too small to reflect the effect of indentation on the flow of AH) to 1.0 mm (too large a volume causes too much pressure in the anterior chamber, which results in biased data).

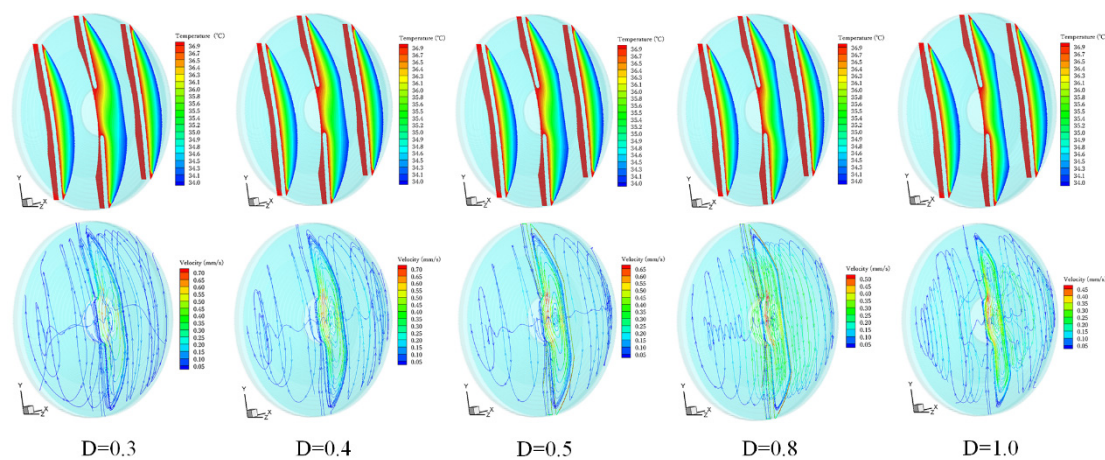


Figure 9. Effect of corneal indentation depth under standing.

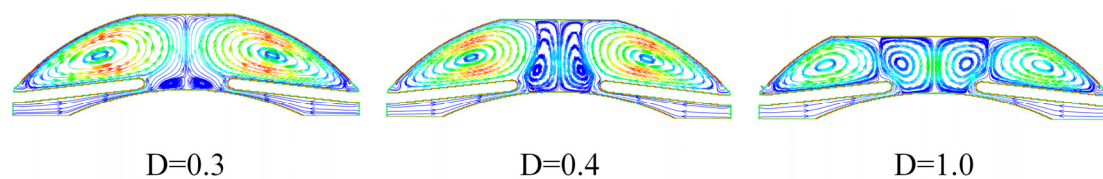


Figure 10. Effect of corneal indentation depth under supine.

The effect of corneal indentation depth on AH flow is given in Figures 9 and 10. Figure 10 gives only the distribution of AH velocity at different corneal indentation depths. As seen in the figure, the deeper the corneal indentation, the weaker the temperature diffusion and the smaller the vortex formed in the AC by the AH flow. In addition, in supine, AH flow formed a secondary vortex in the center of the AC and, as the depth of the corneal indentation increased, the secondary vortex gradually increased. Table 3 gives the magnitude of the maximum velocity of AH at different corneal indentation depths. As can be seen from the table, the deeper the corneal indentation, the smaller the AH velocity. Corneal indentation results in a smaller AC space, weaker convective diffusion of AH,

and, thus, slower AH flow.

Table 3. Effect of corneal indentation on AH maximum velocity.

Depth of corneal indentation (mm)	0	0.3	0.4	0.5	0.8	1.0
Maximum velocity in standing (mm/s)	0.743	0.729	0.707	0.677	0.554	0.463
Maximum velocity in supine (mm/s)	0.119	0.112	0.108	0.104	0.086	0.073

We fitted the relationship between corneal indentation depth and the maximum velocity of AH to obtain the following equation for the relationship.

$$U_{max,stand} = -0.3056d^2 + 0.01685d + 0.74607 \quad (23)$$

$$U_{max,supine} = -0.03117d^2 - 0.01545d + 0.11918, \quad (24)$$

where, $U_{max,stand}$ and $U_{max,supine}$ are the maximum velocity of AH in standing and supine, and d is the depth of the corneal indentation. The fitted functions for the maximum AH velocity are shown in Figure 11, both as open downward quadratic functions.

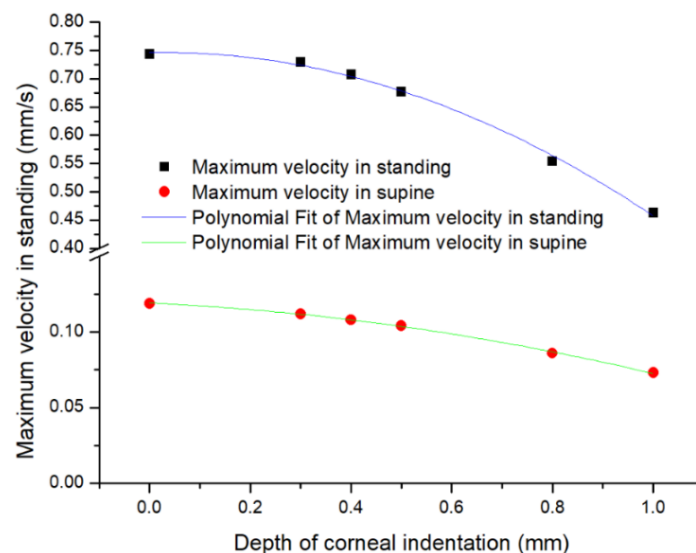


Figure 11. Variation of maximum velocity of AH with corneal indentation depth.

4.5. Effect of the ovoid AC

Normal AH flow simulations use a spherical, symmetrical geometric model of the AC; however, the real eye structure is not perfectly symmetrical. Therefore, we constructed an asymmetric model of the AC to study the effect on AH flow.

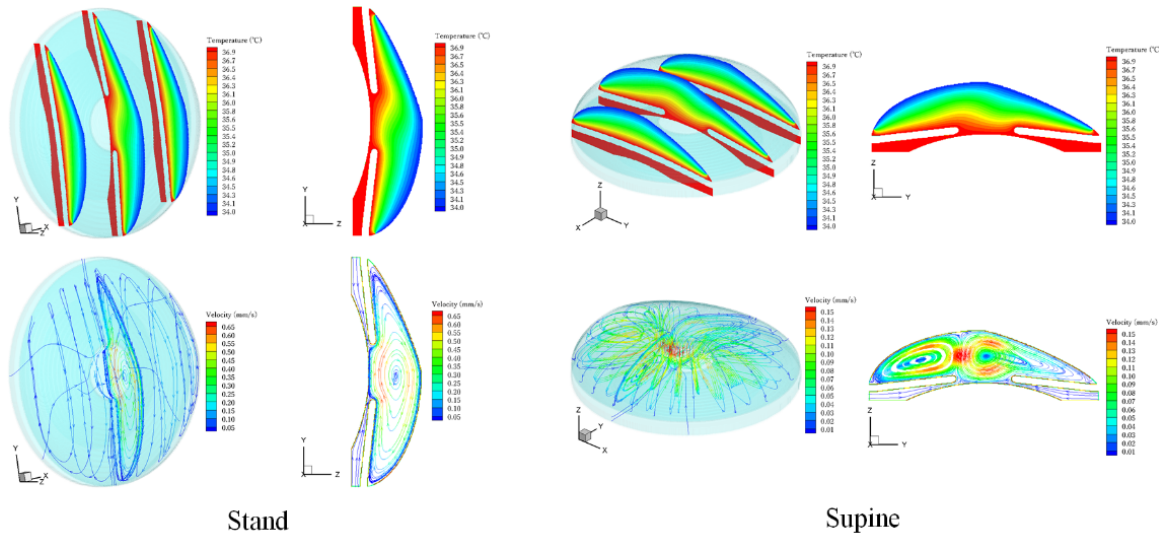


Figure 12. Effect of the ovoid AC on AH flow.

We used an ovoid AC model (one end is circular, and one end is non-circular, in which the non-circular part is constructed in a two-dimensional plane with decreasing slope, and then the two-dimensional plane is used as the basis for rotation to obtain the three-dimensional flow field), and the simulation results are shown in Figure 12. As can be seen from the figure, the ovoid AC has little effect on AH flow in standing; however, it has a significant effect in supine. The asymmetry of the AC causes an asymmetric distribution of AH temperature in the supine position and a more intense diffusion in the AC of the lower hemisphere. Moreover, AH flow forms vortices of varying sizes in the AC, with the larger vortices and maximum velocities in the AC of the lower hemisphere. The maximum velocity of AH in standing and supine was calculated to be 0.655 and 0.160 mm/s, respectively. Thus, it was shown to be 11.84% lower and 34.45% enhanced compared to the maximum velocity of AH in normal eyes (0.743 and 0.119 mm/s), respectively.

5. GPU performance evaluation

The performance of LBM programs is generally expressed using Millions of Lattice Updates Per Second (MLUPS) [22,30], and the larger the MLUPS, the better the performance of the program, which is defined as follows:

$$M_L = \frac{NK_x K_y K_z}{10^6 T_N}, \quad (25)$$

where, N is the number of evolutionary steps, K_i is the number of lattices in the flow field direction and T_N is the evolutionary time.

Table 4. Lab environment.

Device	Version
CPU	Intel(R) Xeon(R) Gold 5117 2.0GHz
GPU	Tesla P100
Operating system	Windows Server 2012
Development tool	Visual Studio 2015
CUDA	V10.0

The lab environment is shown in Table 4. Usually, there are three schemes for dimensional division of threads: 3D Grid with 3D Block [19], 2D Grid with 1D Block [17,18] and 1D Grid with 1D Block [22]. We compute and find the best performance of the 1D Grid and 1D Block [22] scheme when Grid = (276480, 1, 1) and Block = (64, 1, 1). Meanwhile, the shared memory is the on-chip memory of the GPU, which features lower latency and can improve the performance of CUDA programs [18]. We compare the performance of various GPU parallel algorithms on the program by using CPU (sequential single core CPU version) and GPU to evolve the program for 10,000 steps, respectively, and counting the corresponding evolution time, MLUPS and Speedup [19].

Table 5. GPU algorithm performance.

Scheme	Memory (GB)	T_N (s)	MLUPS	Speedup
CPU	7.56	99385.9	1.78	/
Complete	7.56	175.56	1007.89	566.10
Indirect	3.04	185.09	955.99	536.95
Sparse	3.60	145.11	1219.40	684.90
PartSparse	3.57	118.65	1491.29	837.61

The experimental results are shown in Table 5 and Figure 13. As can be seen from the results, the GPU-based LBM algorithm can significantly improve the performance of the program, and all GPU parallel algorithms based on sparse characteristics can save more than 50% of memory. The best performance is the PartSparse scheme, which achieves 1491.29 MLUPS and a Speedup of 837.61 times. The advantages of the PartSparse scheme are obvious, saving a large amount of memory, while increasing the parallelism of thread accesses.

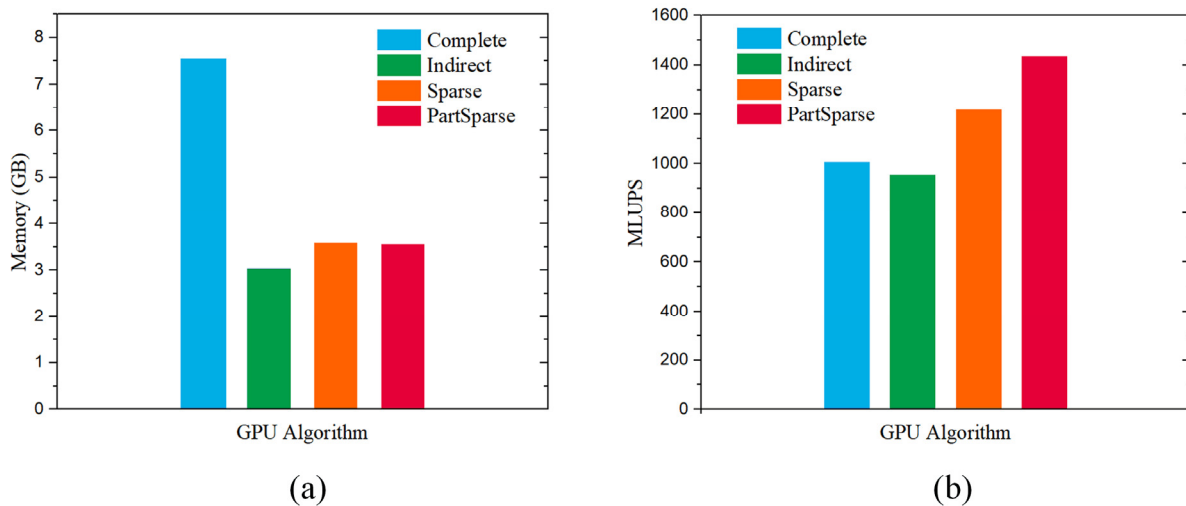


Figure 13. Performance comparison of GPU parallel algorithms.

6. Conclusions

In this paper, we constructed a 3D human eyes anterior segment geometry model, coupled incompressible Navier-Stokes flow, thermal convection and diffusion, and Darcy seepage flow using LBM, and established a 3D human eyes AH dynamics model, which was designed and optimized in parallel by GPU technology. We conclude the following:

1) The AH velocity and temperature distribution under normal intraocular temperature difference were compared with the relevant literature, and the results were in good agreement, which demonstrated the feasibility of the 3D human eyes AH dynamics model.

2) The effects of different factors on AH flow were investigated. The experimental results showed that the AH velocity was 0.05–0.743 mm/s in the standing, and 0.01–0.119 mm/s in the supine position. Additionally, the maximum AH velocity in standing was 6.24 times higher than that in the supine position, indicating that the AH flowed faster in standing; the intraocular temperature difference had the greatest effect on AH flow, indicating that the AH flow was mainly driven by the buoyancy force dominated by the temperature difference; and the AH secretion rate and TM permeability had a greater effect on IOP, where IOP increased with the increase of AH secretion rate, but decreased as the TM permeability increased.

3) The deeper the corneal indentation, the slower the AH flow, and the weaker the convective diffusion. The asymmetric nature of the ovoid AC results in an asymmetric distribution of AH temperature and flow vortex in the supine position.

4) The use of GPU parallelism can significantly improve the performance of the LB algorithm. Our proposed PartSparse algorithm saved more than 50% of memory consumption, and achieved a performance of 1491.29 MLUPS, which can obtain a Speedup of 837.61 times.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 12062005, 81860635, 11862003), the Project of Guangxi Natural Science Foundation (Grant No.

2021JJA110093) and the Key Research Program (Grant No. 2018ZD006) for Guangxi Normal University.

Conflict of interest

We declare that there are no competing interests.

References

1. C. R. Canning, M. J. Greaney, J. N. Dewynne, A. D. Fitt, Fluid flow in the anterior chamber of a human eye, *Math. Med. Biol.*, **19** (2002), 31–60. <https://doi.org/10.1093/imammb/19.1.31>
2. J. J. Heys, V. H. Barocas, A boussinesq model of natural convection in the human eye and the formation of Krukenberg's spindle, *Ann. Biomed. Eng.*, **30** (2002), 392–401. <https://doi.org/10.1114/1.1477447>
3. A. D. Fitt, G. Gonzalez, Fluid mechanics of the human eye: Aqueous humour flow in the anterior chamber, *Bull. Math. Biol.*, **68** (2006), 53–71. <https://doi.org/10.1007/s11538-005-9015-2>
4. T. R. Crowder, V. J. Ervin, Numerical simulations of fluid pressure in the human eye, *Appl. Math. Comput.*, **219** (2013), 11119–11133. <https://doi.org/10.1016/j.amc.2013.04.060>
5. S. Kumar, S. Acharya, R. Beuerman, A. Palkama, Numerical solution of ocular fluid dynamics in a rabbit eye: parametric effects, *Ann. Biomed. Eng.*, **34** (2006), 530. <https://doi.org/10.1007/s10439-005-9048-6>
6. J. A. Ferreira, P. De Oliveira, P. M. Da Silva, J. N. Murta, Numerical simulation of aqueous humor flow: from healthy to pathologic situations, *Appl. Math. Comput.*, **226** (2014), 777–792. <https://doi.org/10.1016/j.amc.2013.10.070>
7. G. R. Mcnamara, G. Zanetti, Use of the Boltzmann equation to simulate lattice-gas automata, *Phys. Rev. Lett.*, **61** (1988), 2332–2335. <https://doi.org/10.1103/PhysRevLett.61.2332>
8. S. Y. Chen, H. D. Chen, D. Martinez, W. Matthaeus, Lattice Boltzmann model for simulation of magnetohydrodynamics, *Phys. Rev. Lett.*, **67** (1991), 3776–3779. <https://doi.org/10.1103/PhysRevLett.67.3776>
9. Y. H. Qian, D. D'Humières, P. Lallemand, Lattice bgk models for navier-stokes equation, *Europhys. Lett.*, **17** (1992), 479–484. <https://doi.org/10.1209/0295-5075/17/6/001>
10. Y. H. Qian, Simulating thermohydrodynamics with lattice BGK models, *J. Sci. Comput.*, **8** (1993), 231–242. <https://doi.org/10.1007/BF01060932>
11. Z. Guo, B. Shi, C. Zheng, A coupled lattice BGK model for the Boussinesq equations, *Int. J. Numer. Methods Fluids*, **39** (2002), 325–342. <https://doi.org/10.1002/flid.337>
12. Z. Guo, T. Zhao, A lattice Boltzmann model for convection heat transfer in porous media, *Numer. Heat Transfer, Part B*, **47** (2005), 157–177. <https://doi.org/10.1080/10407790590883405>
13. Z. Qin, L. Meng, F. Yang, C. Zhang, B. Wen, Aqueous humor dynamics in human eye: a lattice Boltzmann study, *Math. Biosci. Eng.*, **18** (2021), 5006–5028. <https://doi.org/10.3934/mbe.2021255>
14. E. Lindholm, J. Nickolls, S. Oberman, J. Montrym, NVIDIA Tesla: a unified graphics and computing architecture, *IEEE Micro*, **28** (2008), 39–55. <https://doi.org/10.1109/MM.2008.31>

15. J. Tölke, Implementation of a Lattice Boltzmann kernel using the compute unified device architecture developed by Nvidia, *Comput. Visualization Sci.*, **13** (2008), 29–39. <https://doi.org/10.1007/s00791-008-0120-2>
16. F. Kuznik, C. Obrecht, G. Rusaouen, J. J. Roux, LBM based flow simulation using GPU computing processor, *Comput. Math. Appl.*, **59** (2010), 2380–2392. <https://doi.org/10.1016/j.camwa.2009.08.052>
17. J. Habich, T. Zeiser, G. Hager, G. Wellein, Performance analysis and optimization strategies for a D3Q19 lattice Boltzmann kernel on nVIDIA GPUs using CUDA, *Adv. Eng. Software*, **42** (2011), 266–272. <https://doi.org/10.1016/j.advengsoft.2010.10.007>
18. P. R. Rinaldi, E. A. Dari, M. J. Vénere, A. Clause, A lattice-Boltzmann solver for 3D fluid simulation on GPU, *Simul. Modell. Pract. Theory*, **25** (2012), 163–171. <https://doi.org/10.1016/j.simpat.2012.03.004>
19. Z. Wang, Y. Zhao, A. P. Sawchuck, M. C. Dalsing, H. Yu, GPU acceleration of Volumetric Lattice Boltzmann Method for patient-specific computational hemodynamics, *Comput. Fluids*, **115** (2015), 192–200. <https://doi.org/10.1016/j.compfluid.2015.04.004>
20. A. Xu, L. Shi, T. S. Zhao, Accelerated lattice Boltzmann simulation using GPU and OpenACC with data management, *Int. J. Heat Mass Transfer*, **109** (2017), 577–588. <https://doi.org/10.1016/j.ijheatmasstransfer.2017.02.032>
21. B. Ma, L. Shi, C. Huang, Q. Xu, Effects of nanoscale pore structure on permeability and relative permeability loss analyzed by GPU enhanced Multiple-Relaxation-Time LBM, *Int. J. Heat Mass Transfer*, **117** (2018), 584–594. <https://doi.org/10.1016/j.ijheatmasstransfer.2017.09.136>
22. Á. Salinas, C. Torres, O. Ayala, A fast and efficient integration of boundary conditions into a unified CUDA Kernel for a shallow water solver lattice Boltzmann method, *Comput. Phys. Commun.*, **249** (2020). <https://doi.org/10.1016/j.cpc.2019.107009>
23. J. J. Heys, V. H. Barocas, M. J. Taravella, Modeling passive mechanical interaction between aqueous humor and iris, *J. Biomech. Eng.*, **123** (2001), 540–547. <https://doi.org/10.1115/1.1411972>
24. O. Abouali, A. Modareszadeh, A. Ghaffarieh, J. Tu, Investigation of saccadic eye movement effects on the fluid dynamic in the anterior chamber, *J. Biomech. Eng.*, **134** (2012). <https://doi.org/10.1115/1.4005762>
25. C. Lin, F. Yuan, Numerical simulations of ethacrynic acid transport from precorneal region to trabecular meshwork, *Ann. Biomed. Eng.*, **38** (2010), 935–944. <https://doi.org/10.1007/s10439-010-9947-z>
26. N. Heussner, L. Holl, T. Nowak, T. Beuth, M. S. Spitzer, W. Stork, Prediction of temperature and damage in an irradiated human eye—Utilization of a detailed computer model which includes a vectorial blood stream in the choroid, *Comput. Biol. Med.*, **51** (2014), 35–43. <https://doi.org/10.1016/j.compbiomed.2014.04.021>
27. Z. Guo, C. Zheng, B. Shi, Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method, *Chin. Phys.*, **11** (2002), 366. <https://doi.org/10.1088/1009-1963/11/4/310>
28. M. Astorino, J. B. Sagredo, A. Quarteroni, A modular lattice boltzmann solver for GPU computing processors, *SeMA J.*, **59** (2013), 53–78. <https://doi.org/10.1007/BF03322610>

29. C. Nita, L. M. Itu, C. Suci, GPU accelerated blood flow computation using the Lattice Boltzmann method, in *2013 IEEE High Performance Extreme Computing Conference (HPEC)*, IEEE, 2013. <https://doi.org/10.1109/HPEC.2013.6670324>
30. A. G. Shet, S. H. Sorathiya, S. Krithivasan, A. M. Deshpande, B. Kaul, S. D. Sherlekar, et al., Data structure and movement for lattice-based simulations, *Phys. Rev. E*, **88** (2013), 013314. <https://doi.org/10.1103/PhysRevE.88.013314>
31. M. J. Mawson, A. J. Revell, Memory transfer optimization for a lattice Boltzmann solver on Kepler architecture nVidia GPUs, *Comput. Phys. Commun.*, **185** (2014), 2566–2574. <https://doi.org/10.1016/j.cpc.2014.06.003>
32. C. Huang, B. Shi, Z. Guo, Z. Chai, Multi-GPU based Lattice Boltzmann method for hemodynamic simulation in patient-specific cerebral aneurysm, *Commun. Comput. Phys.*, **17** (2015), 960–974. <https://doi.org/10.4208/cicp.2014.m342>
33. A. Karampatzakis, T. Samaras, Numerical model of heat transfer in the human eye with consideration of fluid dynamics of the aqueous humour, *Phys. Med. Biol.*, **55** (2010), 5653–5665. <https://doi.org/10.1088/0031-9155/55/19/003>
34. Y. Zhao, B. Chen, D. Li, Optimization of surgical protocol for laser iridotomy based on the numerical simulation of aqueous flow, *Math. Biosci. Eng.*, **16** (2019), 7405–7420. <https://doi.org/10.3934/mbe.2019370>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)