9th ANKARA INTERNATIONAL AEROSPACE CONFERENCE
20-22 September 2017 - METU, Ankara TURKEY

AIAC-2017-059

CONJUGATE HEAT TRANSFER COMPUTATIONS WITH LATTICE BOLTZMANN METHOD AND COMPARISONS ITS RESULTS WITH FINITE VOLUME METHOD

Gregorio Gerardo Spinelli*
Istanbul Technical University
Istanbul, Turkey

Bayram Celik†
Istanbul Technical University
Istanbul, Turkey

ABSTRACT

We have developed a lattice Boltzmann method solver that is able to model conjugated heat transfer problems for fluid and solid in contact. The solver uses a Cartesian mesh that consists of D_2Q_9 lattices. Transient heat transfer in solid and fluid regions are solved simultaneously without specific treatment on the interface. In order to accelerate the solver we use Compute Unified Device Architecture (CUDA) for coding. We developed another conjugate heat transfer solver that is based on finite volume method by modifying two existing open source CFD program, openFoam. The solver satisfies energy balance and no jump in temperature on the interface with inner iterations. This study compares the two solvers by modeling a benchmark problem, backward facing step duct flow over a thick wall.

INTRODUCTION

Lattice Boltzmann Method (LBM) was first introduced in 1988 by McNamara and Zanetti as a novel approach for the solution of fluid flow problems [McNamara and Zanetti 1988]. Apart from the conventional CFD solvers, the lattice Boltzmann method is based on gas-kinetic theory. The discrete Boltzmann equation is solved on a Cartesian mesh, where each lattice is linked to the neighboring lattices by streaming directions. The most preferred lattice model is D_2Q_9 for 2-D, in which each lattice has 9 streaming directions. Particle collision is modeled as a linearized Bhatnagar-Gross-Krook (BGK) expansion, which is responsible for the local change of the microscopic quantities (particle distribution function). From the hydrodynamic moments of the particle distributions, macroscopic quantities (velocity, pressure, and temperature) are obtained [Perumal and Dass 2015]. Nowadays, there are energy distribution functions based on either temperature or internal energy [Chen, Shu, and Tan 2017; D'Orazio et al. 2015; Karimipour et al. 2012; Monfared et al. 2015], while the model for velocity distribution function is unique [Mohamad 2011; Seddiq, Maerefat, and Mirzaei 2014; Zhao and Yong 2017]. Both distribution functions are extendable to the desired order of accuracy.

A great number of boundary conditions are available today for lattice Boltzmann method solvers [Contrino et al. 2014; Jahanshaloo et al. 2016; Li, Mei, and Klausner 2013; Rahmati, Ashrafizaadeh, and

*PhD Student in Astronautical Engineering Department, Email: spinelli15@itu.edu.tr
†Assistant Professor in Astronautical Engineering Department, Email: celikbay@itu.edu.tr

Table 1: Nomenclature

Symbol	Description	Symbol	Description
Roman		*Greek*	
b	Wall thickness	α	Thermal diffusivity
$\vec{c}$	Lattice velocity	Θ	Dissipation function
c_s	LBM speed of sound	λ	Second coefficient of viscosity
C_v	Specific heat at constant volume	μ	Dynamic viscosity
e	Internal energy	ν	Kinematic viscosity
E	Total energy	ρ	Density
f	Momentum distribution function	τ	Shear stress
$\tilde{f}$	Post-collision f	ω	Collision frequency
$\vec{F}$	Vector of fluxes for FVM	Ω	Control volume
g	Temperature distribution function	$\partial\Omega$	Surface of the control volume
$\tilde{g}$	Post-collision g		
h	Entrance height	*Superscript*	
H	Channel height	eq	Equilibrium value
j	General index	n	Quantity value at time n
k	Thermal conductivity		
k_r	Solid to fluid conductivity ratio	*Subscript*	
L	Characteristic length	c	Conservative quantity
$\vec{n}$	Normal vector of the surface	f	Relative to fluid region
Nu	Nusselt number	i	Along i^{th} direction
p	Pressure	m	Related to momentum equation
Pr	Prandtl number	$\max$	Maximum value
q	Heat flux	$\min$	Minimum value
$\vec{Q}$	Vector of conserved variables	s	Relative to solid region
$\vec{r}$	Position vector	v	Viscous quantity
R	Gas constant	x	Along x direction
Re	Reynolds number	y	Along y direction
S	Edge length for a quad element		
T	Temperature	*Mathematical Signs*	
t	Time	Δ	Increment
u	Horizontal component of $\vec{V}$	∇	Gradient
v	Vertical component of $\vec{V}$		
$\vec{V}$	Velocity vector		
w	Specific weight for lattice		
x	x-axis		
y	y-axis		

Shirani 2014]. More information on boundary conditions can be found in several review articles, such as [Jahanshaloo et al. 2016].

Conjugate heat transfer (CHT) problems require solution of energy equation at once, in the regions that are in contact. Here the regions in contact are fluid and solid. The temperature distribution in fluid, results from both convection and conduction. On the other hand, only conduction is considered in the solid. Therefore, a proper solution for a CHT problem gives contact surface temperature accurately. Most importantly, at the contact surface, the temperature and energy flux shouldn't have any jump [Ramšak 2015].

Lattice Boltzmann method has great a potential to accelerate on Graphics Processing Unit (GPU). We tested our solver performance on GPU by modeling several fluid flow problems [Spinelli and Celik 2015, 2017; Spinelli, Luca, and Celik 2015].

The main objective of this study is to compare the results obtained from CHT LBM solver with those obtained from conventional CFD solvers [Celik 2017].

NUMERICAL METHODS

lattice Boltzmann method

The aforementioned Bhatnagar-Gross- Krook linearized expansion model [Mohamad 2011], simplifies the complicated right hand side of the original Boltzmann equation as below:

$$\frac{\partial f_i}{\partial t} + \vec{c}_i \cdot \vec{\nabla} f_i = \omega(f_i^{\mathrm{eq}} - f_i) \tag{1}$$

Collision, streaming, boundary condition updating, and calculation of macroscopic quantities are the four main steps of LBM;

The D_2Q_9 model is shown in Fig. 1.

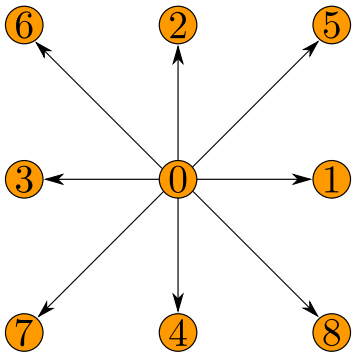

Figure 1: The D_2Q_9 lattice model

Tab. 2 shows weighting factors and local velocities for each streaming direction, for D_2Q_9 model.

Table 2: Weighting factors and local velocity vectors for D_2Q_9 model

i	0	1	2	3	4	5	6	7	8
w_i	$4/9$	$1/9$	$1/9$	$1/9$	$1/9$	$1/36$	$1/36$	$1/36$	$1/36$
$\vec{c}_i$	$(0,0)$	$(1,0)$	$(0,1)$	$(-1,0)$	$(0,-1)$	$(1,1)$	$(-1,1)$	$(-1,-1)$	$(1,-1)$

Discretizing Eq.1 in time and space we obtain

$$f_i(\vec{r} + \vec{c}_i\Delta t, t + \Delta t) = f_i(\vec{r}, t) + \omega_m \left[f_i^{\mathrm{eq}}(\vec{r}, t) - f_i(\vec{r}, t)\right] \tag{2}$$

Streaming and collision stages can be written explicitly as follows:

$$\tilde{f}_i(\vec{r}, t) = f_i(\vec{r}, t) + \omega_m \left[f_i^{\mathrm{eq}}(\vec{r}, t) - f_i(\vec{r}, t)\right] \tag{3}$$

$$f_i(\vec{r} + \vec{c}_i\Delta t, t + \Delta t) = \tilde{f}_i(\vec{r}, t) \tag{4}$$

Coding collision and streaming stages on GPU are shown with two examples in List.'s 1 & 2, respectively.

Listing 1: A sample code for *collision* stage on GPU

```
__global__ void Collision_GPU(type *g, type *T, type *w, type *u, type *
    v, type *cx, type *cy, type omega, int Lx, int Ly, int Lz, type cs,
    type dt)
{
  int j = blockIdx.y * blockDim.y + threadIdx.y;
  int i = blockIdx.x * blockDim.x + threadIdx.x;
```

```
  if ((i < Lx) && (j < Ly))
  {
    for (int k = 0; k < Lz; k++)
    {
      float geq = T[j * Lx + i] * w[k] * (1. + 1./(cs*cs) * (u[j * Lx +
   i] * cx[k] + v[j * Lx + i] * cy[k]));
      g[k * Lx * Ly + j * Lx + i] = (1. - omega) * g[k * Lx * Ly + j *
   Lx + i] + omega * geq;
    }
  }
}
```

Listing 2: A sample code for *streaming* in directions 1 and 3 on GPU

```
__global__ void streaming_GPU(float *f, int Lx, int Ly)
{
  int j = blockIdx.y * blockDim.y + threadIdx.y;
  int i = blockIdx.x * blockDim.x + threadIdx.x;

  //adjust indeces
  int j_adj = (Ly - 1) - j;
  int i_adj = (Lx - 1) - i;

  if ((i < Lx - 1) && (j < Ly))
  {
    // right to left
    f[1 * Lx * Ly + j_adj * Lx + i_adj] = f[1 * Lx * Ly + j_adj * Lx + (
   i_adj - 1)];
    // left to right
    f[3 * Lx * Ly + j * Lx + i] = f[3 * Lx * Ly + j * Lx + (i + 1)];
  }
}
```

Then, macroscopic quantities are calculated as follows:

$$\rho = \sum_{i=1}^{9} f_i \tag{5}$$

$$\vec{V} = \sum_{i=1}^{9} f_i \vec{c}_i \tag{6}$$

A second order accurate velocity model used in the computation is written below:

$$f_i^{\text{eq}} = w_i \rho \left[1 + \frac{\vec{c}_i \cdot \vec{V}}{c_s^2} + 0.5 \frac{(\vec{c}_i \cdot \vec{V})^2}{c_s^4} - 0.5 \frac{\vec{V} \cdot \vec{V}}{c_s^2} \right] \tag{7}$$

The kinematic viscosity ν of the flow is related to the collision frequency ω_m as follow:

$$\nu = \frac{\Delta x^2}{3\Delta t} \left(\omega_m - 0.5 \right) \tag{8}$$

Although the streaming and collision stages of temperature calculation in CHT are almost identical to those for only fluid solver, a few discrepancies in the procedure are listed below.

- additional distribution function, g is introduced for T;
- the macroscopic quantity is T instead of ρ;
- now collision frequency ω it is related to the thermal diffusivity α.

In order to give a better insight a flow chart of the solver is given in Fig. 4, in Appendix.

Finite volume solver

Governing equations for an incompressible, laminar flow of Newtonian fluid are as follows.

$$\vec{\nabla} \cdot \vec{V} = 0 \tag{9}$$

$$\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \vec{\nabla}\vec{V} = \frac{1}{\rho}\nabla p + \nu \nabla^2 \vec{V} \tag{10}$$

$$\frac{\partial T_f}{\partial t} + \vec{V} \cdot \vec{\nabla} T_f = \alpha \nabla^2 T_f \tag{11}$$

Just Eq. 11 needs to e solved for solid region, where $\vec{V} = \vec{0}$.

Two separate solvers of openFoam [Craven and Campbell n.d.; *OpenFOAM The Open Source CFD Toolbox - Programmer's Guide* n.d.], for fluid and solid regions, are combined to generate Finite Volume Method (FVM) based solver. icoFoam is the transient incompressible solver selected for the fluid region, at which the energy equation is added. laplacianFoam solver is modified for solving the conduction equation in solid region. For details see the article [Celik 2017].

RESULTS AND DISCUSSION

We already tested the accuracy of the LBM solver by modeling benchmark problems for fluid flow [Spinelli and Celik 2015, 2017]. In order to see the solver performance on pure conduction, we modeled the problem of conduction on a rectangular plate and compared the result with the analytical solution.

In order to check the accuracy of the solver, two CHT problems are modeled. The accuracy of the LBM solver is tested by modeling two separate problems; a solid plate with walls at constant temperature and a backward facing step duct that has a solid wall with thickness.

Test case #1

The geometry is a simple rectangular plate, with a length of L_x and L_y in x and y directions. The plate boundaries are at 0 temperature, except the one on the top at $T = 1$. The comparison is made by using the analytical solution [Cengel et al. 2003].

Comparison of the obtained temperature distribution along the centerline with analytical solution shows that maximum deviation is less than 0.14%.

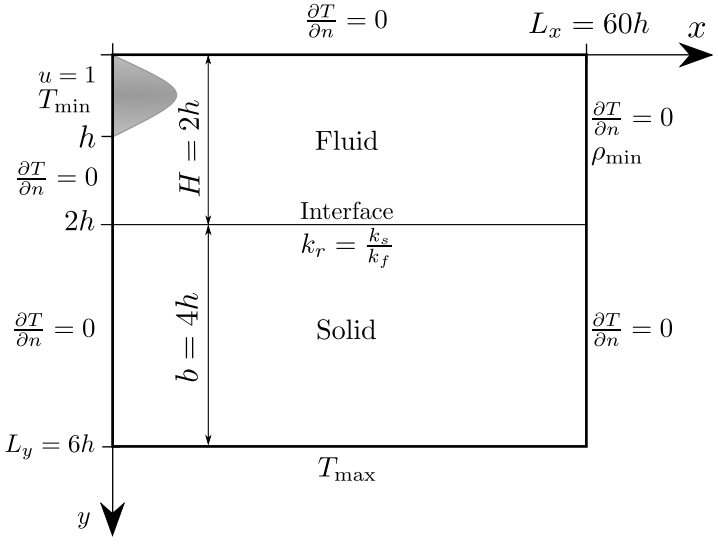

Figure 2: Initial and boundary conditions of Test Case #2

Test case #2

The second problem simulated here is the conjugate heat transfer through a backward facing step duct [Kanna and Das 2006]. A fluid at 0 temperature enters the channel from the left, where the channel is suddenly expanding. As can be seen from Fig. 2, the bottom wall of the channel has a thickness of two times the channel height. It is assumed that the flow entering the channel is fully developed. The solid to fluid conductivity ratio is equal to 10. Except the bottom wall and inlet, all walls are adiabatic. The bottom wall is at $T = 1$.

Accurate determination of interface temperature requires solution of both fluid flow and combined energy equation in two regions. Interface Nusselt number distribution for $\mathrm{Re} = 800$ and solid to fluid k_r ratio of 10 are shown in Fig. 3, Alongside the results available in literature [Ramšak 2015] and the one obtained from FVM solver [Celik 2017]. At the entrance region, the LBM solution is more smooth than FVM solution, the latter presents a sudden drop for interface Nusselt number distribution right after entrance station.

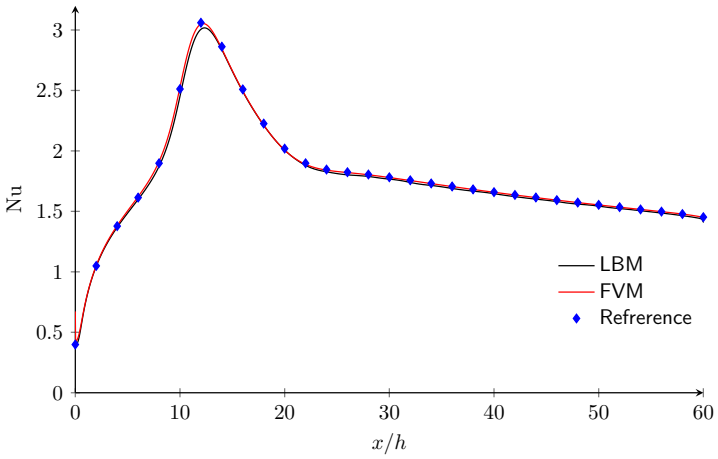

Figure 3: Nusselt number comparison for test case #2, where $\mathrm{Re} = 800, \mathrm{Pr} = 0.71, k_r = 10$

Along with the qualitative comparison in Fig. 3, numerical data are listed in Tab. 3. Despite small oscillatory behavior, the maximum descrepancy between LBM and [Ramšak 2015] is less than 1.41%. On the other hand, the maximum descrepancy between FVM and [Ramšak 2015] is less than 0.37%, if we neglect the entrance region where the error is about 10.95%.

Table 3: Accuracy comparison between LBM and FVM with respect to reference [Ramšak 2015]

$Re = 800, Pr = 0.71, k_r = 10$

Test Case	$T_{\min}$	$T_{\max}$	T_{avg}	$Nu_{\min}$	$Nu_{\max}$	Nu_{avg}
LBM	0.4565	0.8063	0.6362	0.3954	3.0174	1.7752
FVM	0.4592	0.8141	0.6408	0.4477	3.0540	1.7848
Ramsak	0.4640	0.8090	0.6412	0.3987	3.0600	1.7914
% Error LBM-Ramsak	1.6429	0.3349	0.7859	0.8346	1.4118	0.9143
% Error FVM-Ramsak	1.0453	0.6265	0.0624	10.9448	0.1965	0.3698

Furthermore, a comparison between LBM and FVM results is given both in terms of minimum, maximum, and average values of temperature and Nusselt number variation along the interface, in Tab. 4 and Tab. 5 respectively, for different Reynolds numbers, Prandtl numbers and solid-to-fluid thermal conductivity ratios.

For temperature evaluation the maximum deviation observed between LBM and FVM is 2.66%, which occurs for extremum comparisons. If we consider average temperature value at the interface the maximum difference is about 1.43% for cases where $\mathrm{Re} = 200$, while it is below 0.53% for $\mathrm{Re} = 400$ and 0.72% for $\mathrm{Re} = 800$.

For Nusselt number the behavior is slightly different. The maximum difference for extremum is 13.23%. As for temperature case, maximum deviation is 2.24% for average Nusselt number along the interface for $\mathrm{Re} = 200$, while it is less than 0.54% for the other cases.

Table 4: Comparison between LBM and FVM for interface temperature evaluation

Test Case			*LBM*			*% Error LBM-FVM*		
Re	Pr	k_r	$T_{\min}$	$T_{\max}$	T_{avg}	$T_{\min}$	$T_{\max}$	T_{avg}
200	0.10	10	0.7649	0.9963	0.9284	0.7713	0.1405	0.2703
200	0.71	10	0.6071	0.8255	0.7438	1.2189	1.2598	1.3310
200	0.71	50	0.8687	0.9579	0.9279	2.2562	0.8560	1.4226
200	0.71	100	0.9171	0.9753	0.9567	2.6606	0.9330	1.4320
200	10	10	0.3796	0.5862	0.5114	0.5269	2.0641	1.3688
400	0.10	10	0.7362	0.9679	0.8744	0.3939	0.2376	0.2287
400	0.71	10	0.5597	0.7871	0.6914	0.6968	0.4320	0.5351
800	0.71	10	0.4565	0.8063	0.6362	0.5915	0.9674	0.7230

Table 5: Comparison between LBM and FVM for interface Nusselt number evaluation

Test Case			*LBM*			*% Error LBM-FVM*		
Re	Pr	k_r	$Nu_{\min}$	$Nu_{\max}$	Nu_{avg}	$Nu_{\min}$	$Nu_{\max}$	Nu_{avg}
200	0.71	10	0.5797	2.1232	1.2325	9.2117	2.3832	0.0649
200	10	10	0.7281	3.2596	2.3448	12.9790	2.7549	2.2365
400	0.71	10	0.4562	2.3904	1.5084	11.2889	0.9329	0.2824
800	0.71	10	0.3954	3.0174	1.7752	13.2271	1.2130	0.5425

CONCLUSION

This paper provides and compares two solvers for Conjugate Heat Transfer problem: lattice Boltzmann method and finite volume method. The LBM solver that uses Single Relaxation Time model has instability for Reynolds number greater than 2000. Our findings are summarized below:

- The FVM solver is more stable than LBM solver for higher Reynolds numbers.
- The FVM code has less deviation from [Ramšak 2015] results than LBM code, when comparing average values.
- In general the two solvers give qualitatively similar results.
- The maximum deviation between the two interface average temperature values is about 0.72% for $\mathrm{Re} = 800$, 0.53% for $\mathrm{Re} = 400$, and is 1.43% for $\mathrm{Re} = 200$.
- The maximum difference between the two interface extremum temperature values is about 0.97% for $\mathrm{Re} = 800$, 0.70% for $\mathrm{Re} = 400$, and is 2.66% for $\mathrm{Re} = 200$.
- The maximum deviation between the two interface average Nusselt values is about 0.54% for $\mathrm{Re} = 800$, 0.28% for $\mathrm{Re} = 400$, and is 0.065% for $\mathrm{Re} = 200$.
- The maximum difference between the two interface minimum Nusselt values is about 13.23% for $\mathrm{Re} = 800$, 11.29% for $\mathrm{Re} = 400$, and is 12.98% for $\mathrm{Re} = 200$. The deviation between the two interface maximum Nusselt values reflect the same trend of deviation between the two interface average Nusselt values.
- The smallest difference is observed always for low Prandtl numbers, about 0.10.

ACKNOWLEDGEMENTS

Gregorio Gerardo Spinelli would like to thank TUBITAK for Graduate Scholarship Programme for International Students (Code 2215). This study is supported by Istanbul Technical University (BAP project No:36592).

APPENDIX

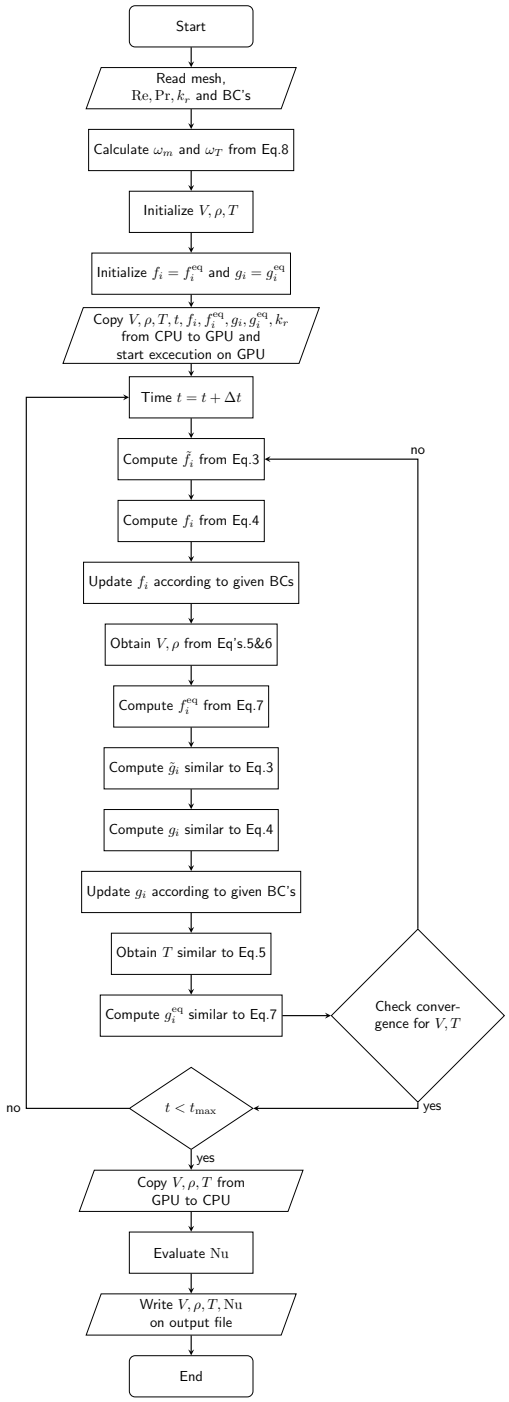

Figure 4: Flow chart of the developed CHT LBM solver

References

Celik, Bayram (2017). "Conjugate heat transfer characteristics of laminar flows through a backward facing step duct". In: *Suleyman Demirel University Journal of Natural and Applied Sciences* (in review).

Cengel, Yunus et al. (2003). "Heat transfer: a practical approach". In: *MacGraw Hill, New York*.

Chen, Z, C Shu, and D Tan (2017). "A simplified thermal lattice Boltzmann method without evolution of distribution functions". In: *International Journal of Heat and Mass Transfer* 105, pp. 741–757.

Contrino, Dario et al. (2014). "Lattice-Boltzmann simulations of the thermally driven 2D square cavity at high Rayleigh numbers". In: *Journal of Computational Physics* 275, pp. 257–272.

Craven, Brent A. and Robert L. Campbell (n.d.). *Multi-Region Conjugate Heat-Mass Transfer*. (Last reached on 27.07.2017). `http://www.personal.psu.edu/dab143/OFW6/Training/craven_slides.pdf`.

D'Orazio, Annunziata et al. (2015). "Lattice Boltzmann method with heat flux boundary condition applied to mixed convection in inclined lid driven cavity". In: *Meccanica* 50.4, pp. 945–962.

Jahanshaloo, Leila et al. (2016). "An overview of boundary implementation in lattice Boltzmann method for computational heat and mass transfer". In: *International Communications in Heat and Mass Transfer* 78, pp. 1–12.

Kanna, P Rajesh and Manab Kumar Das (2006). "Conjugate heat transfer study of backward-facing step flow–a benchmark problem". In: *International journal of heat and mass transfer* 49.21, pp. 3929–3941.

Karimipour, Arash et al. (2012). "Investigation of the gravity effects on the mixed convection heat transfer in a microchannel using lattice Boltzmann method". In: *International Journal of Thermal Sciences* 54, pp. 142–152.

Li, Like, Renwei Mei, and James F Klausner (2013). "Boundary conditions for thermal lattice Boltzmann equation method". In: *Journal of Computational Physics* 237, pp. 366–395.

McNamara, Guy R and Gianluigi Zanetti (1988). "Use of the Boltzmann equation to simulate lattice-gas automata". In: *Physical review letters* 61.20, p. 2332.

Mohamad, Abdulmajeed A (2011). *Lattice Boltzmann method: fundamentals and engineering applications with computer codes*. Springer Science & Business Media.

Monfared, Amir Ehsan Feili et al. (2015). "Thermal flux simulations by lattice Boltzmann method; investigation of high Richardson number cross flows over tandem square cylinders". In: *International Journal of Heat and Mass Transfer* 86, pp. 563–580.

OpenFOAM The Open Source CFD Toolbox - Programmer's Guide (n.d.). (Last reached on 27.07.2017). `http://foam.sourceforge.net/docs/Guides-a4/ProgrammersGuide.pdf`.

Perumal, D Arumuga and Anoop K Dass (2015). "A Review on the development of lattice Boltzmann computation of macro fluid flows and heat transfer". In: *Alexandria Engineering Journal* 54.4, pp. 955–971.

Rahmati, AR, M Ashrafizaadeh, and E Shirani (2014). "A Multi-Relaxation-Time Lattice Boltzmann Method on Non-Uniform Grids for Large Eddy Simulation of Rayleigh-Bénard Convection Using Two Sub-Grid Scale Models." In: *Journal of Applied Fluid Mechanics* 7.1.

Ramšak, Matjaž (2015). "Conjugate heat transfer of backward-facing step flow: A benchmark problem revisited". In: *International Journal of Heat and Mass Transfer* 84, pp. 791–799.

Seddiq, Mehdi, Mehdi Maerefat, and Masaud Mirzaei (2014). "Modeling of heat transfer at the fluid–solid interface by lattice Boltzmann method". In: *International Journal of Thermal Sciences* 75, pp. 28–35.

Spinelli, Gregorio Gerardo and Bayram Celik (2015). "GPU Accelarated multi-block lattice Boltzmann solver for viscous flow problems". In: *Ankara International Aerospace Conference VIII* (in proceedings).

— (2017). "Conjugate heat transfer computations via lattice boltzmann method". In: *Journal of Aeronautics and Space Technologies* (in press).

Spinelli, Gregorio Gerardo, Luigi de Luca, and Bayram Celik (2015). "GPU accelarated multi-block lattice boltzmann solver for viscous flow problems". MA thesis. University of Naples "Federico II".

Zhao, Weifeng and Wen-An Yong (2017). "Single-node second-order boundary schemes for the lattice Boltzmann method". In: *Journal of Computational Physics* 329, pp. 1–15.